CENA 20.000 zł

ISSN 0867-3918 INDEKS 377112

64 PLUS 4 & AMIGA

9-10/'92

MIESIĘCZNIK UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE



Informujemy, że nasze pismo można w dalszym ciągu **zaprenumerować** - co daje pewność systematycznego otrzymywania (drogą pocztową). Nasz miesięcznik kosztuje w prenumeracie 10.000 zł. Prenumeratę można zawrzeć na okres nie krótszy niż dwa miesiące, w dowolnym okresie, maksymalnie do końca roku kalendarzowego. Wykupujący prenumeratę nie ponoszą kosztów przesyłki pocztowej.

OD REDAKCJI

Przedsiębiorstwo ABUK S-ka z o.o. oferuje państwu **szybką i tanią obsługę reklamową.** Ogłoszenia drobne od osób indywidualnych (do 10 słów) przyjmujemy bezpłatnie. Większe - 1000 zł za słowo. Reklamy ramkowe (minimalny format - 20 cm^2): 1cm^2 ogłoszenia **-8000zł, cała strona - 3,0 mln zł**; każdy kolor - odpowiednio 100% drożej. Ogłoszenia przyjmujemy za pośrednictwem poczty (nasz adres - patrz stopka redakcyjna). Treść ogłoszenia z określeniem formatu reklamy (ewentualnie zamówieniem koloru) prosimy nadsyłać listem poleconym wraz z odcinkiem wpłaty. Wpłat prosimy dokonywać za pomocą przekazu pieniężnego na konto Przedsiębiorstwa ABUK, Bank Polska Kasa Opieki SA Oddział w Bydgoszczy, konto nr : 5.09011-400522.7-2511-30-111.0. Dołączenie do zamówienia odcinka wpłaty przyspieszy zamieszczenie reklamy. Redakcja nie ponosi odpowiedzialności za treść i wiarygodność ogłoszeń.

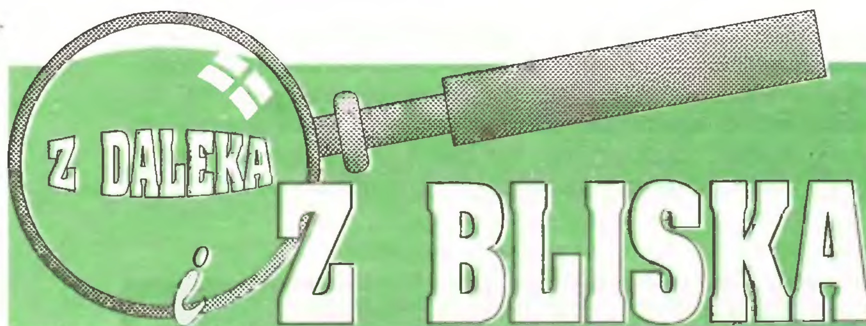


miesięcznik nr 9,10(22)
wrzesień/listopad 1992
cena 1 egz.: 10.000 zł.
64PLUS4

WYDAWCA: ABUK Spółka z o.o.
REDAGUJĄ: Waldemar Szczygieł (redaktor naczelny) z zespołem.
ADRES REDAKCJI: Redakcja „64 plus 4", 85-166 Bydgoszcz 43, skrytka pocztowa 64.
OKŁADKA: Piotr Bartz.
SKŁAD: ABUK
DRUK: POLRASTER, Bydgoszcz.

W numerze :

NOWINKI



PARTY

W dniach od 28 do 30 sierpnia odbyło się w Głogowie trzecie już spotkanie użytkowników komputerów C64 - organizatorami była znana grupa Asphyxia. Na party zjawiło się sporo osób znanych, a także wiele reprezentujących nowe polskie grupy. Z tych ważniejszych warto wymienić: Skylight, Elysium, Axel, Crazy Boys, Flash Inc., Triad, Fenix, Charged, Lords Of Darkness, Inflexion, Brains, Nebula, Taboo oraz Parados.

Party zaczęło się w piątek o godzinie dziesiątej w Miejskim Ośrodku Kultury w Głogowie. Duża sala przeznaczona była dla użytkowników C64 oraz druga, znacznie mniejsza dla Amigowców. Dla tych którzy nie zmieścili się w salach pozostawał jeszcze główny hol budynku. Organizatorzy przygotowali także pokój sypialny (mimo to sporo osób spało w śpiworach rozłożonych we wszystkich kątach sali).

Rankiem w sobotę około godziny siódmej wszyscy zaczęli się powoli budzić i tak rozpoczynał się drugi dzień spotkania. Część uczestników zabrała się za dokańczanie swoich prac przeznaczonych do wystawiania w konkursach o najlepsze demo, muzykę oraz grafikę stworzoną na Commodore 64. Pozostali w większości zajmowali się masowym kopiowaniem nowych programów. Wielu wymieniało ze sobą adresy zawiązując w ten sposób nowe kontakty.

W miarę zbliżania się wieczoru wszystkie grupy przygotowujące swoje produkcje na konkurs poprawiały ostatnie znalezione błędy doprowadzając swoje dzieła do niemalże doskonałości. W konkursie brały udział następujące grupy: Skylight, Inflexion, Charged, Fenix oraz Elysium. Swoje demo pokazała także grupa Asphyxia, która jednak nie brała udziału w konkursie. Najlepszym okazało się demo grupy Elysium, które prawdopodobnie będzie umieszczone na jednym z najbliższych Public Domain Packów.

Podsumowując muszę stwierdzić, że impreza była udana, chociaż ciągle nie dorównywała zlotom użytkowników Amigi. Być może jednak już wkrótce "scena" Commodore 64 dorówna tej Amigowej...

Amiga 4000

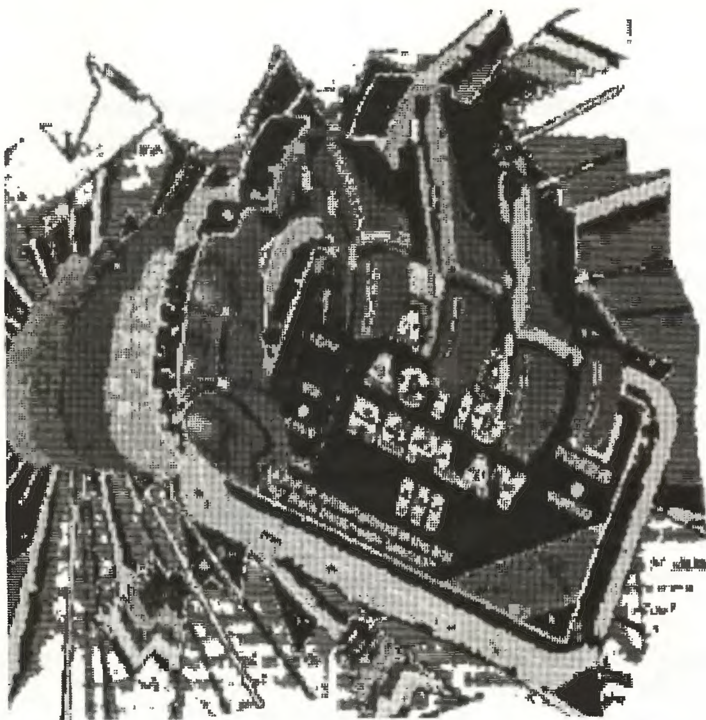
Ostatnio w niemalże wszystkich BBSach w Europie Zachodniej pojawił się tekst zawierający informacje na temat nowego produktu firmy Commodore - Amigi 4000. Informacje podane w dalszej części artykułu nie są być może w pełni zgodne z tym co zaprezentuje wkrótce Commodore, jednak publikujemy je jako ciekawostkę, z którą warto się zapoznać.

Nowa Amiga ma być oparta na zestawie kości nazwanym AA. Jej projektanci mają nadzieję, że będzie to maszyna mogąca swobodnie stawić czoła innym wysokiej klasy komputerom znajdującym się na rynku. A4000 będzie wyposażona w procesor 68040 oraz nowy zestaw chipów ("AA"). Pamięc RAM typu fast będzie rozszerzalna do 64MB, natomiast pamięć typu chip do 16MB. Prawdopodobnie rozszerzanie pamięci w końcu odbywać się bedzie za pomocą pakietów typu SIMM. Oczywiście A4000 będzie standardowo wyposażana w twardy dysk. Stacje dysków montowane w A4000 bedą o dużej gęstości zapisu, dzięki czemu jedna dyskietka HD będzie mogła zawierać aż 1.76MB danych. Porty rozszerzenia pozostają takie jak w A3000 (Zorro III).

Drugi model Amigi, jaki ma zostać zaprezentowany będzie umieszczony w nieco mniejszym pudełku o wysokości 3 cale oraz szerokości 15 cali. Będzie wyposażony w procesor 68030 taktowany zegarem 16MHz. Oczywiście zastosowane w nim będą także nowe chipy oraz zamontowane stacje dysków o zwiekszonej gęstości zapisu. Porty rozszerzenia i komunikacja z otoczeniem - podobne do A3000. W obu modelach wyprowadzone są wszystkie standardowe wyjścia oraz kilka dodatkowych portów: dla monitora oraz do podłączenie mikrofonu pracującego z wbudowanym fabrycznie samplerem (umożliwiającym sampling z częstotliwością 22kHz i zapis na 8 bitach). Nowością bedzie zainstalowanie gniazd midi. Mysz niestety pozostaje ciągle ta sama, co nie jest najlepszym pomysłem ze strony Commodore.

Teraz kilka słów o nowym zestawie chipów. Graficzne możliwości, dzięki ich wprowadzeniu, zostaną nieco rozbudowane. Dojdzie kilka ciekawych rozdzielczości: 800 na 300, 800 na 600 (interlace), 1280 na 480 oraz 1280 na 960 (interlace). Chip odpowiedzialny za likwidowanie interlace (tzw. flicker--fixer) może współpracować ze wszystkimi z podanych rozdzielczości, jednakże nie każdy monitor VGA będzie mógł utrzymać rodzielczość pionową 960 pikseli. Każdy z trybów może pracować nawet na 8 bitplaneach, wyjątkiem jest tryb 1280, w którym możliwe jest wykorzystanie tylko 6 bitplane'ów. Paleta kolorów jest już oczywiście 24 bitowa, istnieje także możlwość zasymulowania palety 12 bitowej, aby zachować zgodność ze starymi programami. Oczywiście zachowany jest także tryb HAM w każdej z rodzielczości. Poprawiono także wykorzystanie sprite'ów, które zmieniają się w zależności od używanej rozdzielczości. Powoduje to między innymi także zmiany kształtu strzałki kursora. Sprite'y mogą mieć szerokość maksymalnie 64 piksele, natomiast ich wysokość nie jest w żaden sposób limitowana. Niestety w jednej linii nie może być ich nigdy więcej niż osiem, brak ten może chociaż częściowo zrekompensuje możliwość wykorzystania w nich do 16 kolorów. To już wszystko, co znaleźliśmy dla waş na temat nowości z Commodore. Mamy nadzieję, że inforamcje te choć w minimalnej części okażą się prawdziwe, gdyż komputery te mogą stanowić dobrą konkurencję dla komputerów PC oraz Maców...

J. H.



AMIGA ACTION REPLAY

MK III

wersja do:
Amigi 500/plus
tylko 1550tys.
Amigi 2000 1720tys.

DODATKOWA MOC DLA TWOJEJ AMIGI

Teraz z 256kb własnego systemu, jeszcze więcej możliwosci niż w MK II

- ® kompletny M68000 assembler i disassembler
- ® instrukcje do tworzenia niesmiertelności
- ® instrukcje do wyciągania i obróbki grafiki
- ® pełna kontrola antywirusowa
- ® bootselector
- ® wszystkie instrukcje monitora dysku
- ® wybór PAL-NTSC
- ® najpotrzebniejsze instrukcje Amiga-DOS
- ® instrukcje do wyciągania muzyki
- ® doskonały program kopiujący BurstNibbler
- ® możliwość ustalania dowolnej konfiguracji pamięci i stacji dysków
- ® możliwość oglądania i modyfikacji wszystkich rejestrów (także Read-only)

Te i masą innych możliwości otrzymujesz w każdej chwili po nacisnięciu przycisku Freeze w Action'ie.
Niezbędna pomoc dla każdego kodera, hackera, muzyka, grafika i potencjalnego gracza

Wyłącznym dystrybutorem Action Replay'a w Polsce jest:

EUREKA SOFT & HARDWARE

tel./fax.(066)-362-072 ul. Żwirki i Wigury 13a 62-300 Września

Czekamy na zamówienia hurtowe. Atrakcyjne upusty!
Ponadto oferujemy:

- rozszerzenia pamięci do 1MB Golden Image (4 kości, zegar) tylko 550tys.
- stacje dysków 3.5 - 1150tys. 5.25" - 1300tys.
- myszki Fancy Mouse tylko 200tys.
- rozszerzenie pamięci do 2MB do Amigi 500 plus - 940tys.
- oprogramowanie Public Domain i Shareware. Proszę przesłać 20tys. po katalog.
W sprzedaży hurtowej - duże ilosci dyskietek Noname i firmowych na cele zaopatrzeniowo-inwestycyjne, boxy na dyskietki, mouse pady

Zamów listownie lub telefonicznie, koszty przesyłki za pobraniem ok. 6% wartości.

C-64

ACIK POCZĄTKUJĄCEGO GRAFIKA

Jak zapowiadałem w poprzednim odcinku naszego cyklu, dzisiaj zajmiemy się rysowaniem na ekranie kolejnej figury geometrycznej. Mowa oczywiście o okręgu. Nie bedziemy zajmować się rysowaniem figur typu prostokąt, czy kwadrat gdyż w bardzo prosty sposob można te figury narysować przy pomocy samych tylko linii. Okrąg natomiast wymaga zastosowania kilku niezbędnych wzorów. W drugiej części artykułu, także przy pomocy samego tylko podprogramu rysowania punktu nauczymy się rysować wykresy niektórych funkcji matematycznych dwuwymiarowych.

Zacznijmy jednak od okręgu. Opisany on jest równaniem:

X^2+Y^2=R^2

gdzie X oraz Y to współrzędne punktu spełniającego równanie okręgu, a R to promień danego okręgu. W równaniu tym zawarta jest także informacja, iż środek okręgu, o którym mowa, umieszczony jest w punkcie (0;0), czyli w początku układu współrzędnych. Korzystając z tego wzoru okrąg o zadanym promieniu można narysować analizując kolejne punkty ekranu oraz sprawdzając, czy ich współrzędne należą do okręgu. Taki sposób jest jednak ogromnie czasochłonny. Należy więc znaleźć inny algorytm - szybszy (na tyle, na ile pozwala programowanie w języku Basic) i dokładniejszy. Możemy jednak zająć się przekształcaniem równania tak, aby otrzymać z niego Y. Po kilku prostych przekształceniach otrzymamy następujące równanie:

Y=SQR (R^2-X^2)

Oczywiście, aby równanie to miało sens musi być spełniona zależność:

-R <= X <= R

Mając aż tyle informacji jesteśmy już właściwie w stanie narysować połowę okręgu (korzystając właśnie z tego wzoru), drugą połowę możemy z powodzeniem rysować korzystając z symetrii osiowej względem osi X. Aby jeszcze przyspieszyć proces rysowania możemy posłużyć się także symetrią osiową względem osi Y. W ten sposób otrzymamy okrąg składający się z czterech identycznych, równolegle rysowanych części. Oto program realizujący opisany algorytm:

```
10 GOSUB 1000
20 GOSUB 1060
30 XS=160: YS=100: R=10
40 X1=0: Y1=R
50 GOSUB 1090
60 X1=X1+0.5: Y1=SQR (R*R-X*X)
70 X=XS+X1: Y=YS+Y1: GOSUB 1090
80 X=XS+X1: Y=YS-Y1: GOSUB 1090
90 X=XS-X1: Y=YS+Y1: GOSUB 1090
100 X=XS-X1: Y=YS-Y1: GOSUB 1090
110 IF X1<R THEN GOTO 50
```

Korzystając jednak z pewnych własności okręgu i punktów należących do niego możemy pokusić się o przyspieszenie rysowania. Nasz nowy algorytm będzie sprawdzał dla zadanego okręgu dwa punkty, które mogą spełniać jego równanie (pracujemy tylko na liczbach całkowitych), następnie wybieramy ten punkt, który jest bliższy pełnej z nim zgodności. Oto moja propozycja nowego programu, który tym razem dostosujemy do potrzeb naszej biblioteki podprogramów graficznych:

```
1119 REM rysowanie okregu (A;B;R)
1120 X1=R: Y1=0
1130 X=A+X1: Y=B+Y1: GOSUB 1090
1140 X=A-X1: Y=B+Y1: GOSUB 1090
1150 X=A+X1: Y=B-Y1: GOSUB 1090
1160 X=A-X1: Y=B-Y1: GOSUB 1090
1170 X=A+Y1: Y=B+X1: GOSUB 1090
1180 X=A+Y1: Y=B-X1: GOSUB 1090
1190 X=A-Y1: Y=B+X1: GOSUB 1090
1200 X=A-Y1: Y=B-X1: GOSUB 1090
1210 OG=MO+Y+Y+1
1220 OU=OG-X-X+1
1230 Y=Y+1: MO=OG
1240 IF ABS(OU) < ABS(OG) THEN X=X-1: MO=OU
1250 IF X1>=Y1 THEN 1130
1260 RETURN
```

Tak oto wyglądać będzie kolejny podprogram w naszym pakiecie graficznym.

Zanim przystąpimy do czytania dalszej części artykułu proponuję najpierw zapoznać się dokładniej z podprogramem rysującym koło i spróbować wykorzystać go w praktyce, na przykład tworząc przy jego pomocy symetryczne wzorki składające się z samych tylko okręgów.

Umiemy już rysować okręgi, weźmy się więc za bardziej skomplikowane figury będą to wykresy funkcji. Zwróćmy uwagę,

C-64

iż pierwszy z algorytmów rysowania okręgu był niczym innym jak tylko rysowainem wykresu pewnej funkcji, a następnie jej odbicia w osi X oraz Y. Podobnie pierwszy algorytm na rysowanie odcinka łączącego dwa różne punkty opierał się na rysowaniu wykresu pewnej funkcji. Jak jednak zabrać się za rysowanie dowolnie wybranych funkcji? Szczerze mówiąc napisanie dobrego programu realizującego to zadanie nie jest wcale takie proste. Zwykle w programach rysujących wykresy podajemy zakres osi X, to znaczy współrzędną początkową oraz końcową wykresu, a program za nas oblicza odpowiednie zakresy na osi Y i rysuje najbardziej optymalny obraz wykresu funkcji. Istnieje także możliwość padawania obu zakresów. Wtedy program umieszcza odpowiednio wykres na ekranie w danym zakresie. My jednakże pozwolimy sobie na wprowadzenie pewnych uproszczeń. Pierwszym, i najważniejszmym z nich, jest przyjęcie, iż żadna z podanych funkcji nie będzie funkcją ciągłą i każda wartość tej funkcji dla podanej dziedziny będzie skończona. Dzięki niemu możemy pominąć wszelkiego rodzaju sprawdzania kolejno rysowanych punktów. Jednak zwrócić musimy uwagę na rysowaną funkcję, aby nie spowodowała ona błędu podczas wykonywania programu. Przed przystąpieniem do sporządzania wykresu mając wartość początkową oraz końcową X musimy znaleźć zakres danych na osi Y.

Następnie wystarczy już tylko znaleźć odpowiednią skale, z której skorzystamy przy zamianie współrzędnych matematycznych na współrzędne ekranowe. Ze względu na konieczność zachowania zwiezłości progamu nie będziemy rysować ani osi układu współrzędnych ani wypisywać na ekranie żadnych innych dodatkowych inforamcji. Za to zapraszam czytelników do pracy nad podanym programem. Oto i nasz program rysujący zadany wykres funkcji:

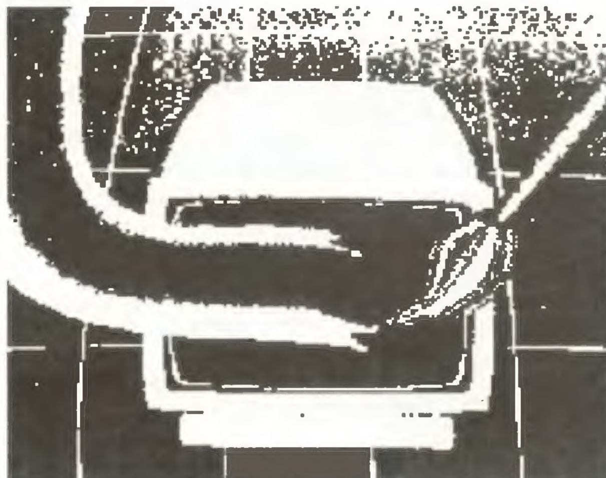
```
1 DEF FN F(X)=SIN(X)
10 GOSUB 1000
20 GOSUB 1060
30 X1=-3.14: X2=3.14
40 K=(X2-X1)/320
45 Y1=FN(X1): Y2=FN(X1)
50 FOR X=X1 TO X2 STEP K
60 Y=FN(X)
70 IF Y2<Y THEN Y2=Y
80 IF Y1>Y THEN Y1=Y
90 NEXT X
100 KX=319/(X2-X1)
110 KY=199/(Y2-Y1)
120 FOR I=X1 TO XG STEP DX
130 X=KX*(I-X1): Y=KY*(Y2-FN(I))
140 GOSUB 1090
150 NEXT I
```

Program ten narysuje na ekranie wykres funkcji sinus. Oczywiście możemy narysować wykres dowolnej innej funkcji pod warunkiem, iż będzie ona spełniała założenia podane we wstępnym opisie programu.

Może teraz kilka słów więcej o działaniu programu. W pierwszej jego linii definiujemy swoją własną funkcje. Kolejne dwie linie: 10 oraz 20 to odwołania do włączenia trybu graficznego, a następnie wyczyszczenia gotowego do pracy ekranu graficznego. W linii 30 definiujemy wartość początkową i końcową dla osi X. Linia 40 to obliczenie odpowiedniego kroku dla pętli analizującej wszystkie wartości zadanej funkcji w podanym zakresie. Kolejne linie od 45 do 90 to właśnie pętla, która znajduje najmniejszą oraz największą wartość funkcji w zadanym przedziale. Linie 100 oraz 110 obliczają współczynnik skali, dzięki któremu możliwe będzie przedstawienie wykresu na ekranie. Kolejne linie programu od 120 do 150 to główna pętla zajmująca się rysowaniem wykresu na ekranie.

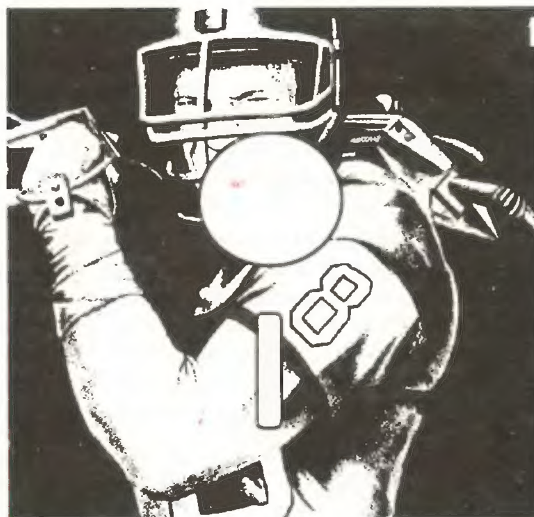
To już wszytko w dzisiejszym odcinku. Za miesiąc chyba w końcu pozbędziemy się naszego pakietu procedur graficznych napisanego w języku Basic i spróbujemy zrealizować podobne procedury w języku maszynowym.

Jarosław "Jarri" Horodecki



C-64

DYSK cz. 3



EX-DOS

Jak zapowiedziałem ostatnio - tematem dzisiejszego odcinka będzie współpraca z dyskiem z poziomu języka Basic. Nie chodzi mi oczywiście o wykonywanie podstawowych rozkazów jak: formatowanie dyskietki, kasowanie pliku czy też standardowe wczytywanie katalogu, ale o bardziej skomplikowane operacje na dyskietce wymagające wczytania do pamięci oraz odpowiedniej obróbki konkretnego sektora.

Aby móc rozpocząć prace z Basic'iem i stacją dysków musimy najpierw nauczyć się wykonywania operacji na dyskietce, to znaczy musimy umieć wczytać dowolny sektor z dyskietki, a następnie zapisać go ponownie na dysku w to samo miejsce. Zadanie - to w brew pozorom - jest bardzo proste i wymagać od nas bedzie napisania dwóch krótkich podprogramów, które będą realizowały te dwa zadania.

Zanim przystąpimy do pisania programu musimy jednak zapoznać się z kilkoma niezbędnymi komendami. Pierwsza z nich to B-R, czyli Block Read (z ang. czytaj blok). Powoduje ona odczytanie z dysku konkretnego sektora oraz zapisanie go w buforze dysku. Składnia tej komedny jest następująca:

```
PRINT#nr,"B-R:"CH;DV;TR;SC
```

gdzie CH oznacza numer kanału, przez który później będą transmitowane dane do komputera (nie jest to określenie w pełni ścisłe, ale na razie możemy tak założyć), DV to numer napędu obsługiwanego przez sterownik, a TR i SC to kolejno numer ścieżki i sektora, z którego ma się odbywać odczyt danych.

Drugą ważną komendą jest B-W, czyli Block Write (zapisz blok). Składnia jej jest identyczna z B-R, tak więc nie będę jej omawiał. Ostanim już potrzebnym nam rozkazem jest B-P, czyli Buffer Pointer. Składnia tego rozkazu różni się od podanych dwóch rozkazów i jest następująca:

```
PRINT#nr,"B-P:"CH;PS
```

gdzie CH jest odpowiednim numerem kanału (tak jak w B-R i B-W), a PS jest pozycją bufora dysku od której zaczynamy pobieranie danych.

Mając te informacje możemy już przystąpić do pisania naszych podprogramów. Pozostało jeszcze zastanowić się, w jakiej formie chcemy przechowywać w pamięci kolejne bajty sektora. Mamy kilka możliwości. Możemy użyć do tego celu Basic'owych struktur danych, czyli tablic. Używanie ich jednak wiąże się z możliwością utraty danych w momencie przerwania pracy programu, poza tym trudno jest nagrać tego rodzaju dane na dysku (tylko jako plik sekwencyjny). Moim zdaniem najwygodniejszym rozwiązaniem będzie wykorzystanie pamięci od adresu 49152, która nie jest używana przez Basic. Możemy wykorzystać przecież 256 pierwszych bajtów na nasze dane. Dodatkowo będziemy mogli łatwo zmieniać zawartość dowolnych komórek oraz przerywać działanie programu nie ryzykując utraty danych.

Oprócz zapamiętywania danych niezbędną rzeczą, którą należy wykonać jest konieczność otwarcia dwóch kanałów dostępu do dysków. Jeden będzie służył do wydawania komend (kanał 15), a drugi do pobierania danych.

Oto moja propozycja wykonania tych czynności:

```
10  OPEN #15,8,15,"I0"
20  OPEN #2,8,2 ,"#"
999 END
```

Od linii 1000 zaczniemy pisać nasze podprogramy. Najpierw oczywiście czytanie z dysku. Przyjmijmy, że numer ścieżki oraz sektora będą podane jako zmienne T oraz S. Oto podprogram, który odczyta odpowiedni sektor i przepisze jego zawartość do pamięci od adresu 49152:

```
1000 PRINT#15,"B-R:";2;0;T;S
1010 PRINT#15,"B-P:";2;0
1020 FOR I=0 TO 255
1030 GET #2,A$: IF A$="" THEN A$=CHR$(0)
1040 POKE49152+I,ASC(A$)
1050 NEXT I
1060 RETURN
```

Podprogram zapisu na dysk będzie bardzo podobny do podanego podprogramu odczytu z dysku. Oczywiście zapisywać będziemy dane umieszczone od adresu 49152:

```
1100 PRINT#15,"B-P:";2;0
1110 FOR I=0 TO 255
1120 A$=CHR$(PEEK(49152+I))
1130 PRINT#2,A$;
1050 NEXT I
1060 PRINT#15,"B-R:";2;0;T;S
1070 RETURN
```

Mamy więc już dwie podstawowe procedury. Są one jednak przydatne tylko w przypadku konieczności działania na całym katalogu dyskietki. My jednak na razie zajmiemy się tylko niektórymi bajtami sektora 0 na ścieżce 18. Aby wykonywać na nich odpowiednie operacje, wystarczy przekazywać dane do bufora dyskietki sterownika stacji dysków. Dzięki temu nie musimy wczytywać do pamięci całej zawartości danego sektora, ale tylko kilka potrzebnych nam bajtów. Dostęp do pojedynczych bajtów otrzymamy oczywiście za pośrednictwem komendy B-P. Drugi jej parametr oznacza właśnie numer bajtu danego sektora, do którego chcemy uzyskać dostęp.

W ten prosty sposób możemy pracować jedynie na pamięci sterownika. Aby wyjaśnić działanie tego rozkazu proponuję posłużyć się dość prostym przykładem.

Jak zapewne pamiętamy z poprzedniego odcinka trzeci

bajt sektora 0 na ścieżce 18 jest odpowiedzialny za możliwość zapisu na dysk oraz kasowania plików z dysku. Dotychczas tego rodzaju zabezpieczenie musieliśmy robić korzystając z monitora dyskowego lub innego tego rodzaju programu. Teraz jednak możemy skorzystać z możliwości, jakie niesie ze sobą pośrednictwo języka Basic. Oto program realizujący to zadanie:

```
10   OPEN 15,8,15 ,"I0"
20   OPEN 2,8,2,"#"
30   PRINT #15,"B-R:";2;0;18;0
40   PRINT #15,"B-P:";2;2
50   PRINT #2,CHR$(255);
60   PRINT #15,"B-W:";2;0 ;18;0
70   CLOSE 2
80 PRINT #15,"I0":CLOSE15
```

Program ten jak widać jest bardzo prosty. Dwie pierwsze linie to oczywiście otwarcie obydwu kanałów do transmisji. Linia 30 to wysłanie do kontolera komendy odczytu ścieżki 18 sektora 0 i wstawienie go do bufora. Następnie, w linii 40 ustawiamy znacznik bufora na pozyjce numer 2 (zaczynamy liczyć od 0 i pomijamy dwa pierwsze bajty). Teraz, w linii 50 wysyłamy nową wartość, która ma być wstawiona na zaznaczone przy pomocy B-P miejsce. W końcu wysyłamy komendę zapisu tego wszystkiego ponownie na dysk oraz inicjujemy stację i zamykamy wszystkie kanały.

Warto zwrócić uwagę na, to aby przed uruchomieniem programu włożyć do stacji dyskietkę, którą chcemy zabezpieczyć. Należało by także zapamiętać, iż zabezpieczona w ten sposób dyskietka może być odbezpieczona tylko i wyłącznie przez formatowanie z indeksem lub użycie specjalnego programu korzystającego z własnych procedur odczytu sektora.

Pozostańmy jeszcze przez chwilę w tym samym sektorze. Jako kolejny przykład proponuje napisanie programiku, który umożliwi nam wpisanie pięcioliterowego identyfikatora dyskietki. Zadanie to jest właściwie równie proste jak zabezpieczenie dyskietki, jednak dla naszej wygody będziemy odczytywali także dotychczas używany identyfikator. Indeks dyskietki znajduję się zaraz za jej nazwą, poczynając od 162 pozycji sektora 0 ścieżki 18. Będąc w posiadaniu tej informacji powinniśmy bez żadnych problemów napisać potrzebny program. Mimo to jednak zaprezentuję jeszcze moją propozycję tegoż programu:

```
10   OPEN 15,8,15,"I0"
20   OPEN 2,8,2,"#"
30   PRINT #15,"B-R:";2;0;18;0
40   PRINT #15,"B-P:";2;162
50   FOR I=1 TO 5
60   GET #2,A$: IF A$="" THEN A$=CHR$(0)
70   ID$=ID$+A$
80   NEXT I
90   PRINT "DOTYCHCZASOWE ID: ";ID$
100  INPUT "NOWE ID: ";ID$
110  PRINT #15,"B-P:";2;162
120  PRINT #2,ID$;
130  PRINT #15,"B-W:";2;0;18;0
140  CLOSE 2
150  PRINT #15,"I0" CLOSE 15
```

Program ten, także wymaga od nas dyskietki włożonej do stacji przed jego uruchomieniem. Należy także zwrócić uwagę, aby wprowadzany przez nas nowy identyfikator nie miał większej długości niż pięć znaków.

Do obu prezentowanych programików możemy oczywiście dorobić szereg udogodnień, a także połączyć je w pakiet, dopisać podprogramy edycji katalogu korzystając z podanych procedur odczytu z dysku i stworzyć własny edytor do wprowadzania zmian w katalogu dyskietki.

Po zapoznaniu się z przekazanym w trzech odcinkach naszego cyklu materiałem z powodzeniem można zabrać się za tego rodzaju robotę, oczywiście pod warunkiem, iż dość dobrze znamy język Basic.

Jarosław "Jarri" Horodecki

Nowa taśma

C-64



PUBLIC DOMAIN

DARKSIDE #04 - grupy PARADOS
SID'N'VIC #01
STYLE #01 - grupy SKYLIGHT
CIABACH #04/05 - grupy CRAZY BOYS
HIGHLIFE #011 - grupy ELISIUM
AXEL NEW #005 - grupy AXEL
PEZ IN MY BRAIN - demo grupy Panoramic Design
3X3 Editor - edytor czcionek grupy Skylight
The Addy Corner - magazyn z adresami
Handy Term V.8.4 - terminal
Humor Basic
Gandlaf Protector - do zabezpieczania programów
M&C Editor - do edycji komunikatów i komend z ROM'u
Last Work 4 Asx - trzyczęściowe demo grupy Asphyxia

O zasadach nabywania naszych PDP piszemy na str. 22.

OPIS ZESTAWU PUBLIC DOMAIN PACK

nr 20 (dysk - sierpień '92)

C-64

Oto mamy przed sobą sierpniowe wydanie naszego PDP. Tym razem prezentujemy nieco mniej niż zwykle programów użytkowych, a wiecej demonstracyjnych. Wynika to z faktu, iż podczas wakcji dużo mniej ludzi zajmuje się komputerami, tak więc powstaje znacznie mniej nowych programów użytkowych. Mimo to udało mi się jeszcze znaleźć kilka niezłych programów demonstracyjnych oraz nowy i bardzo wygodny program służący do tworzenia obrazków z wykorzystaniem techniki FLI.

Na początek może kilka słów o edytorze. Jest to kolejny niezły produkt węgierskiej grupy FACES. W odróżnieniu od niektórych innych edytorów tego typu ten posiada dużo przydatnych opcji. Mamy możliwość zapamiętywania ekranu w pamięci i późniejszego odtworzenia (przydatne przy praktycznym sprawdzaniu rożnych elementów tworzonego obrazu, w razie konieczności usunięcia go wystarczy powrócić do poprzedniej wersji). Sama obsługa programu także jest dość wygodna. Na przykład szybkość ruchu koursora może być zmieniana w zależności od upodobań użytkownika. Mamy też możliwość nakładania na tworzony obrazek kraty, która może okazać się pomocna przy dokładnym obliczaniu odległości pomiędzy pojedyńczymi punktami. Możemy także (niespotykane dotąd w innych edytorach) rysować linie, korzystać z narzędzia o nazwie spray. Jest też możliwość rysowania wielokątów wypełnionych lub nie wypełnianych, pudełek. Oczwiście programiści nie zapomnieli także o okręgach. Każdą z narysowanych figur jak i dowolną narysowaną powierzchnię można oczywiście wypełniać. Oprócz wymienionych opcji program posiada także szereg przydatnych funkcji edycyjnych. Wszystkie one są dostępne przez naciśnięcie odpowiednich klawiszy. Autorzy umieścili dokładny opis działania całej klawiatury - dostępny po naciśnięciu klawisza DEL.

Podsumowując, mogę stwierdzić, iż jest to z pewnością najlepszy edytor do tworzenia grafiki FLI dostępny w tej chwili na rynku.

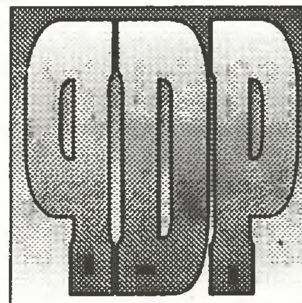
Po krótkim opisie programu użytkowego przyszedł czas na kilka słów o demach znajdujących się na naszym packu. Pierwsze z nich to demo grupy LIGHT pod tytułem BRUTALITY. Składa się on z sześciu części oraz intra używanego ostanio przez grupę LIGHT we wszystkich jej produkcjach. Pierwsza część demonstracji to dwa ciekawe efekty. Pierwszy z nich to trzy niezależne scrolle wykonane w technice FLI. Oprócz nich na ekranie znajduje się dość ładnie narysowany napis "light" oraz multiplexer. Po pewnym czasie na ekranie pokazuje się kilka dość małych obrazków przedstawiających fraktale, czyli figury spełniające pewne matematyczne własności (może w przyszłości powiemy trochę wiecej na ten temat na łamach 64+4 & Amiga). Druga część dema to ponownie bardzo starannie wykonana grafika i niezły pomysł. Efekt zastosowany w tej cześci można nazwać ray-tracingiem, jednak nie jest to ray-tracing w pełnym tego słowa znaczeniu. Widzimy kule poruszającą się po różnych torach z odbijającą się w niej trójwymiarową szachownicą. Nad tą animacją porusza się całkiem ładny scroll sinusowy oraz multiplexer. Tor ruchu kulki można zmieniać przez naciśnięcie klaiwsza "commodore". Komputer wylicza wtedy kolejne klatki nowej animacji.

Kolejna, trzecia część demonstracji to oczywiście kolejny nowy efekt. Mamy tym razem okazję obejrzeć pierwsze na C64 wypełniane wektory z cieniem. Dostępnych jest kilka obiektów jednak mimo dość skomplikowanego kodu, część ta nie robi dużego wrażenia na oglądającym. Czwarta część dema to już dość stary efekt. Logo wykonane w technice FLI wyginające się w prawo oraz w lewo (tzw. TECH-TECH) o cztery kolumny. Grafika jednak jak zwykle stoi na dość wysokim poziomie. Oprócz głównego efektu widzimy także scrolle poruszające się nad obrazkiem wykonanym w trybie multicolor. Piąta część zawiera animację, tzw. ray-tracing. Animacje te najczęściej tworzone są na Amidze, a następnie przenoszone przy pomocy specjalnego programu na C64. Ostania część tego dema to ponownie grafika wektorowa, tym razem jednak nie wypełniana. Część ta zawiera tzw. TV-BOX, czyli sześcian, na którego ścianach znajdują się inne figury wektorowe lub różnego rodzaju animacje. Ze względu na konieczność szybkiego wykonania dość dużej ilości obliczeń matematycznych wykonanie takiego efektu z pewnością nie należy do najłatwiejszych. Nie wspomniałem jeszcze o muzyce w demonstracji, na którą z pewnością warto zwrócić uwagę. Podsumowując demo to jest napewno jedną z najlepszych produkcji ostatniego okresu czasu i warto dołączyć je do swojej kolekcji.

Drugie demo to produkcja grupy TRINOMIC pod tytułem Krombacher. Demo zaczyna się efektownie wykonaną czołówką z bardzo ładnie wykonaną grafiką. Pierwsza część nie zawiera żadnych skomplikowanych efektów, jednak jest starannie zaprojektowana, a grafika stoi na dość wysokim poziomie. Druga część to bardzo ładne logo wychodzące na boczną ramkę oraz scroll obwijający się wokół drugiego scrolla. Całość mimo dość łatwego kodu wygląda bardzo ładnie. Kolejna część to znany już z innych programów demonstracyjnych efekt plazmy. Mamy możliwość zmiany dwóch parametrów opisujących ruch kolorowych kwadratów, przez co możemy wpływać na uzyskiwane efekty. Na plazmę nałożony jest dodatkowo scroll poruszający się w osi Y. W następnej cześci możemy obejrzeć kilka dość ciekawych efektów uzyskanych dzięki użyciu procedury FPP, która daje możliwość umieszczenia dowolnej linii ekranu w dowolnym jego miejscu. Efektów jest dość dużo, tak więc aby je wszystkie obejrzeć część tą należy długo oglądać.

Kolejna część to dość ładny obrazek i właściwie nic więcej, gdyż dwóch dość prostych scrolli nie można nazwać efektem.

Ostatnia już część to procedura łącząca procedurą A-FLI oraz FPP. Widzimy zwykły trójkolorowy scroll przesuwający się nad kolorowymi wzorkami. Efekt ten jest dość popularny i ostanio dość często wykorzystywany w wielu demonstracjach. Szkoda, że autorzy nie wykorzystali w pełni możliwości tej procedury.



PUBLIC DOMAIN PACK

nr 21 (wrzesień '92)

C-64

Po nieco uboższym pod wzgledęm programów użytkowych PDP sierpniowym, wrześniowy PDP prezentuje się już znacznie lepiej. Zamieściliśmy na nim kilka ciekawych programów użytkowych, które zostały napisane przez naszych rodzimych programistów, jest także kilka dobrych programów napisanych przez programistów zachodnich. Nie zapomnieliśmy także o miłośnikach polskich magazynów dyskowych. Zamieściliśmy pierwsze numery kolejnych dwóch magazynów, których wydawanie rozpoczęły raczej nieznane i jeszcze stosunkowo młode, polskie grupy. W końcu oczywiście prezentujemy także najnowsze i chyba obecnie najlepsze na C64 programy demonstracyjne. Przyjrzyjmy się jednak tym programom nieco dokładniej.

Na początek kilka słów o zamieszczonych użytkach: BAR EDITOR V1.0 Program ten został napisany, aby ułatwić życie wszystkim, którzy dotychczas tworząc swoje własne programy demonstracyjne wykorzystujące efekt "tęczy" musieli spędzać dużo czasu nad prawidłowym i eleganckim ustawieniem kolorów kolejnych rastrów. Dzięki programowi BAR EDIOR V1.0 czynność ta staje się o wiele prostsza. W wygodny sposób można ustawić wszystkie kolory na dość dużym obszarze w ciągu kilku minut. Mamy do swojej dyspozycji wiele ciekawych i przydatnych opcji jak: kopiowanie pewnych określonych obszarów w inne miejsce, a także kasowanie tychże obszarów. Także edycja samych kolorów jest bardzo wygodna. Może się ona odbywać zarówno za pomocą klawiszy od '0' do 'F' odpowiadającym kolejnym kolorom, jak i za pomocą '<' oraz '>' przez "przewijanie" kolejnych kolorów. Oczywiście możemy naszą pracę w dowolnym momencie zgrać na dysk, można także wczytywać już gotowe kolory przeznaczone tylko do poprawki.

DIGI EDITOR V1.0. Program ten z pewnością okaże się bardzo przydatnym dla wszystkich zajmujących się muzyką i wykorzystaniem sampli. Umożliwia on przeprowadzenie dość wygodnej edycji dowolnych sampli o dowolnej długości. Z ciekawszych opcji należałoby wymienić możliwość łączenia dwóch sampli w jeden, dzięki czemu mamy wrażenie odtwarzania dwóch niezależnych sampli w tym samym momencie.

ZAK RIPPER V2.0 - kolejny i trzeba przyznać, że również całkiem niezły program użytkowy stworzony przez członków grupy SUN. Umożliwia on bardzo proste i - co ważniejsze - szybkie wyciągniecie niemal dowolnej muzyczki z większości programów demonstracyjnych (z niektórymi, w bardziej skomplikowany sposób zapisanymi, muzyczkami program sobie jednak nie radzi). Po znalezieniu muzyczka jest odtwarzana, a użytkownik w tym czasie może przeszukiwać całą zawartość pamięci (w poszukiwaniu np. danych dla equalizera). Oczywiście program wyświetla także ilość czasu zajmowanego przez procedurę (w rastrach) oraz szereg innych przydatnych informacji. Wyciągniętą muzyczkę można oczywiście zgrać na dysk lub na taśmę.

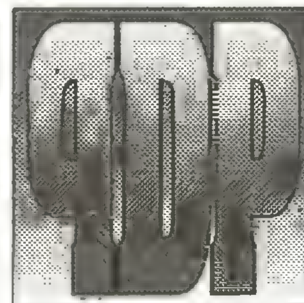
ZIP/UNZIP - kompresor służący do pakowania całych stron dysków. Nie jest to typowy kompresor, gdyż spakowane nim dane można odczytywać tylko z pomocą nigo samego. Do czego więc może być przydaatny tego rodzaju program? Najważniejszym sposobem jego wykorzystania jest pakowanie programów demonstracyjnych, gier i innych zajmujących całe strony dysku. Następnie w ten sposób spakowane programy wysyłane są do BBS'ów (o BBS'ach była już mowa na łamach naszego pisma). Oczywiście dla tego rodzaju kompresora można znaleźć szereg innych zastosowań, jak na przykład przechowywanie rzadko używanych danych (oszczędność dysków!). Program jest bardzo prosty w obsłudze. Po uruchomieniu wystarczy wybrać, czy chcemy pakować (ZIP), czy rozpakowywać (UNZIP). Następnie należy odpowiednio ustawić napęd źródłowy i przeznaczenia i można przystąpić do pracy.

BONKER V1.5 - to zwykły kompresor jednoprzebiegowy. Zamieściliśmy go jednak, ponieważ jest on jednym z najbardziej wydajnych w swojej klasie. Radzimy wypróbować! Obsługa jest bardzo prosta, wystarczy podać nazwę pliku źródłowego, jego adres startu, nazwę pliku docelowego oraz stan komórki $01 odpowiedzialnej za włączanie oraz wyłączanie ROM'ów.

ULTIMATE LYNX Jest to druga wersja znanego już naszym czytelnikom programu LYNX, służącego on do łączenia kilku plików w jeden, ale w takiej formie, iż aby mogły one zostać użyte muszą być ponownie rozpakowane. Program ten jest właściwie przydatny tylko i wyłącznie użytkownikom modemów, którzy dzwonią do różnych BBS'ów.

DIGITAL PAINT V2.0. Program ten jest dość orginalnym programem graficznym. Został stworzony z myślą o obsługujących BBSy działające na C64, a służy do tworzenia grafiki, która może być wyświetlana podczas użytkowania ich systemów. Dzięki temu programowi możemy tworzyć wielokolorowe obrazki lub efektowne ekrany z tekstem składające się tylko i wyłącznie ze standardowych znaków generatora. Oczywiście w ten sposób stworzone obrazy mogą być użyte także we własnych programach, które dzięki temu mimo pracy w trybie tekstowym będą mogły wyglądać naprawdę o wiele ciekawiej. To już wszystkie użytki.

Następnie zamieściliśmy już drugi numer wrocławskiego magazynu po tytułem SID'N'VIC. Magazyn ten znany jest już naszym czytelnikom, więc nie będziemy go opisywać. Warto natomiast zauważyć, iż drugi jego numer w porównaniu z pierwszy jest nieco lepszy. Nie chodzi o sam wygląd magazynu, gdyż ten właściwie się nie zmienił, jednak łatwo jest zauważyć wyższy poziom prezentowanych w kolejnym numerze tekstów.



Kolejnym magazynem zamieszczonym na PDP jest pierwszy numer magazynu NO PROBLEM wydawanego Rockiego i Toma. Najpierw może kilka słów o samym wyglądzie magazynu. Czołówka jest wykonana dość pomysłowo, jednak na pewno mogłaby wyglądać dużo lepiej gdyby



autorzy postarali się o lepszej jakości oprawę graficzną. Poza tym jest jeszcze kilka błędów, które warto było poprawić przed rozprowadzaniem magazynu. Kod samego magazynu nie jest zły. Nie wprowadza jednakże żadnych nowych pomysłów. Artykuły wybieramy z menu, a tekst przesuwany jest za pomocą joysticka w górę i w dół, szybkość przesuwania można zwiększyć za pomocą przycisku FIRE. Niestety także i tu widać niezbyt starannie dopracowaną grafikę. Z pewnością mogłoby być dużo lepiej. Co do poziomu tekstów trudno jest w tej chwili cokolwiek oceniać, gdyż nie jest ich dużo. Pomysły jednak nie są niczym nowym. Możemy znaleźć coś o kodowaniu, także kilka słów o grach oraz opis sceny w Giżycku (mieście, w którym został stworzony magazyn). Na koniec autorzy prezentują swoje adresy. Podsumowując należałoby stwierdzić, iż magazyn ten po wzbogaceniu go o lepszą szatę graficzną mógłby być o wiele lepszy. Przydałoby się także więcej artykułów oraz więcej dobrych pomysłów.

Ostanim już rodzajem programów jakie zostały umieszczone na naszym PDP są programy demonstracyjne. Na ten PDP wybraliśmy kilka najlepszych naszym zdaniem programów z ostatniego miesiąca. Pierwszym z nich jest ostatnie już demo na C64 grupy Horizon (wygrało konkurs na najlepszy tego rodzaju program na party grupy Brutal w Danii). Demo to składa się tylko z trzech części i czołówki, która oprócz bardzo starannego dopracowania nie prezentuje żadnych specjalnie trudnych efektów. Jednak pierwsza część dema jest już z pewnością warta obejrzenia. Widzimy tzw. ray-tracing, czyli animację, której kolejne klatki zostały stworzone dzięki przeprowadzonym odpowiednim obliczeniom matematycznym, w tym także śledzeniu promieni świetlnych (stąd nazwa ray-tracing). Wydawałoby się to dość proste jednak na ekranie w jednym momencie widzimy aż 7 odcieni szarości...

Druga część jest chyba tą, dzięki której grupa Horizon wygrała konkurs. Dwóch koderów z tej grupy, Boogaloo i Exilon, prezentuje w niej bardzo trudne do wykonania na C64 ruchome obiekty wektorowe z wypełnianiem. Oczywiście wszystko jest liczone w czasie rzeczywistym. Oprócz zwykłych obiektów jak sześcian, prosty statek kosmiczny i kilku innych, możemy obejrzeć także samotransformujący się obiekt składający się z 24 prześwitujących ścian. Oprócz obiektów wypełnianych jest też kilka zwykłych wektorów, jednak bardzo ciekawy (i bardzo trudny do wykonania) jest sposób zmiany jednego obiektu w drugi. To trzeba obejrzeć! Ostania część to właściwie zwykły scroll do góry, ale pracujący w trybie Interlace... Nie trzeba dodawać, że całe demo jest idealnie dopracowane pod względem grafiki oraz muzyki, które bardzo dobrze pasują do kolejnych części.

Drugi program demonstracyjny to produkcja jeszcze mało znanej grupy Visual Reality pod tytułem Visuality II. Jest to jedna z pierwszych demonstracji na C64 wykonana w stylu Amigi. Oczywiście mamy okazję obejrzeć sporo dobrych i - co ważniejsze - nowych pomysłów. Są oczywiście dość specyficzne wypełniane obiekty wektorowe. Bardzo ciekawy jest wypełniany TV-BOX oraz oświetlany sześcian. Na koniec autorzy pokazali bardzo ładny obrazek o wielkości dwóch ekranów wykonany w technice interlace poruszający się na boki. Całość jest bardzo starannie wykonana i podobnie jak demo grupy Horizon zasługuje na uwagę.

Jarosław "Jarri" Horodecki

112 SPRITE'ÓW W C-64

C-64

Wiadomo, że standardowo można w C-64 przedstawić 8 sprite'ów. Nasza procedura umożliwia uzyskanie 112 sprite'ów w matrycy 8*14 znaków. Przez wysyłkę przełączania sprite'ów przez przerwania rasterowych linii można wytworzyć do 14 linii z każdą po 8 sprite'ów. A więc wpisujemy listing i wywołujemy instrukcją SYS 8263. Listing drugi pokazuje kod źródłowy.

LISTING 1

```
nazwa: 112 sprite'ow   2000 212d
---------------------------------------------------------------
2000 : 00 00 00 7f ff fe 7f ff e6
2008 : fe 7f ff fe 7f ff fe 7f 98
2010 : ff fe 7f ff fe 7f ff fe 58
2018 : 7f ff fe 7f ff fe 7f ff 3c
2020 : fe 7f ff fe 7f ff fe 7f b0
2028 : ff fe 7f ff fe 7f ff fe 70
2030 : 7f ff fe 7f ff fe 7f ff 54
2038 : fe 7f ff fe 00 00 00 0b ec
2040 : 0b 0c 0f 0f 0c 0b 0b 78 2d
2048 : a9 7f 8d 0d dc a9 f1 8d b4
2050 : 1a d0 a9 00 8d 12 d0 8d 04
2058 : 20 d0 8d 21 d0 a9 1b 8d 4a
2060 : 11 d0 a9 20 8d 15 03 a9 29
2068 : 80 8d 10 d0 a9 ff 8d 15 c8
2070 : d0 20 44 e5 20 66 e5 a2 30
2078 : 00 a9 80 9d f8 07 bd 3f 5e
2080 : 20 9d 27 d0 e8 e0 08 d0 aa
2088 : f0 a9 01 8d 86 02 a9 97 8d
2090 : 8d 14 03 58 60 ea ea ce 9a
2098 : 19 d0 a9 f4 8d 12 d0 a9 22
20a0 : 1b 8d 11 d0 a2 20 ca d0 d8
20a8 : fd a9 13 8d 11 d0 a2 00 12
20b0 : bd fd 20 9d 00 d0 e8 e0 14
20b8 : 10 d0 f5 a2 a0 ca d0 fd a2
20c0 : a2 00 bd 0d 21 9d 00 d0 14
20c8 : e8 e0 10 d0 f5 a2 ff ca 48
20d0 : d0 fd a2 00 bd 1d 21 9d cc
20d8 : 00 d0 e8 e0 10 d0 f5 a0 37
20e0 : 0b a2 8b ca d0 fd a2 01 02
20e8 : bd 00 d0 18 69 15 9d 00 92
20f0 : d0 e8 e8 e0 11 d0 f1 88 fb
20f8 : d0 e7 4c 31 ea 58 01 70 4b
2100 : 01 88 01 a0 01 b8 01 d0 15
2108 : 01 e8 01 00 01 58 16 70 ca
2110 : 16 88 16 a0 16 b8 16 d0 25
2118 : 16 e8 16 00 16 58 05 70 41
2120 : 05 88 05 a0 05 b8 05 d0 8b
2128 : 05 e8 05 00 05 3d 24 44 36
```

LISTING 2

```
290 - .          ba$2000
300 -            ; dane sprite'ow
310 -12000.      by $00,$00,$00,$7f,$ff,$fe,$7f,$ff
330 -            by $fe,$7f,$ff,$fe,$7f,$ff,$fe,$7f
350 -            by $ff,$fe,$7f,$ff,$fe,$7f,$ff,$fe
370 -            by $7f,$ff,$fe,$7f,$ff,$fe,$7f,$ff
390 -            by $fe,$7f,$ff,$fe,$7f,$ff,$fe,$7f
410 -            by $ff,$fe,$7f,$ff,$fe,$7f,$ff,$fe
430 -            by $7f,$ff,$fe,$7f,$ff,$fe,$7f,$ff
450 -            by $fe,$7f,$ff,$fe,$00,$00,$00
460 -            ;kolory sprite'ow
470 -1203f       by $0b,$0b,$0c,$0f,$0f,$0c,$0b,$0b
480 -            ;procedura oprawy
490 - 12047 sei    blokada przerwan
500 -            lda #$7f
510 -            sta $dc0d     ;skasowanie przerwan CIA
520 -            lda #$f1      ;przerwanie linii rastrowych
530 -            sta $d01a     ;wstawienie
540 -            lda #$00
550 -            sta $d012     ;wstawienie linii rastrowych
560 -            sta $d020     ;ramka : czarna
570 -            sta $d021     ;podklad : czarny
580 -            lda #$1b
590 -            sta $d011     ;ekran tekstowy
600 -            lda #$20      ;wstawienie
610 -            sta $0315     ;wektora IRQ(starszego)
620 -            lda #$80      ;pozycja sprite'ow
630 -            sta $d010     ;(bit 8)
640 -            lda #$ff
650 -            sta $d015     wlaczenie sprite'ow
660 -            jsr $e544     skasowanie ekranu
670 -            jsr $e566     ;kursor w gore
680 -            ldx #$00
690 -12079       lda #$80      ;wstawienie
700 -            sta $07f8,x   ;wskaznika sprite'ow
710 -            lda 1203f,x   ;wstawienie
720 -            sta $d027,x   kolorow sprite'ow
730 -            inx
740 -            cpx #$08      ;dla 8 sprite'ow
750 -            bne 12079
760 -            lda #$01      ;kolor pisma : bialy
770 -            sta $0286
780 -            lda #$97      ;wstawienie
790 -            sta $0314     wektora IRQ (mlodszego)
800 -            cli           ;dopuszczenie przerwan
810 -            rts
820 -            nop           ;puste bajty dla JMP
830 -            nop
840 -            ;procedura przewan
850 -            dec $d019     ;skasowanie flagi IRQ
860 -            lda #$f4      ;wstawienie
870 -            sta $d012     ;nowych linii rastrowych
880 -            lda #$1b      ;zmniejszenie ekranu
890 -            sta $d011     na 24 linie
```

C-64

```
900 -          ldx #$20
910 - 120a6    dex           ;opoznienie
920 -          bne 120a6
930 -          lda #$13      ;powiekszenie ekranu
940 -          sta $d011     na 25 linii
950 -          ldx #$00
960 - 120b0    lda 120fd,x   ;wstawienie
970 -          sta $d000,x   ;pozycji sprite'ow
980 -          inx
990 -          cpx #$10      IA;16 bajtow
1000 -         bne 120b0
1010 -         ldx #$a0
1020 - 120bd   dex           ;opoznienie
1030 -         bne 120bd
1040 -         ldx #$00
1050 - 120c2   lda 1210d,x   ;wstawienie
1060 -         sta $d000,x   ;pozycji sprite'ow
1070 -         inx
1080 -         cpx #$10      ;16 bajtow
1090 -         bne 120c2
1100 -         ldx #$ff
1110 - 120cf   dex           ;opoznienie
1120 -         bne 120cf
1130 -         ldx #$00
1140 - 120d4   lda 1211d,x   ;wstawienie
1150 -         sta $d000,x   ;pozycji sprite'ow
1160 -         inx
1170 -         cpx #$10
1180 -         bne 120d4
1190 -         ldy #$0b      ;11 rzedow sprite'ow
1200 - 120e1   ldx #$8b
1210 - 120e3   dex
1220 -         bne 120e3
1230 -         ldx #$01      ;wstawienie licznika
1240 - 120e8   lda $d000,x   ;odczyt wskaznika sprite'ow
1250 -         clc
1260 -         adc #$15      ;dodanie 21
1270 -         sta $d000,x   ;i wstawienie nowego
1280 -         inx
1290 -         inx
1300 -         cpx #$11      ;8 sprite'ow
1310 -         bne 120e8
1320 -         dey
1330 -         bne 120e1
1340 -         jmp $ea31     ;procedura normalnego IRQ
1350 -         ;pozycje sprite'ow
1360 - 120fd   by $58,$01,$70,$01,$88,$01,$a0,$01
1380 -         by $b8,$01,$d0,$01,$e8,$01,$00,$01
1400 - 1210d   by $58,$16,$70,$16,$88,$16,$a0,$16
1420 -         by $b8,$16,$d0,$16,$e8,$16,$00,$16
1440 - 1211d   by $58,$05,$70,$05,$88,$05,$a0,$05
1460 -         by $b8,$05,$d0,$05,$e8,$05,$00,$05
```

opr. R.G.

Rozszerzenie grafiki dla ekranu małej rozdzielczości

Poniższy program -"Basic $B000" przesuwa interpreter Basic'a z $a000-$BFFF na $B000-$CFFF oraz zmienia wszystkie parametry, wartości i adresy.

Basic zamelduje się teraz ilością 43007 zamiast 38911 wolnych bajtów. Zaletą tej metody w stosunku do rozszerzeń RAM'ów jest to, że pamięć jest to dyspozycji w jednym obszarze.

Dodatkowe 4kB można użyć w grach typu adventure względnie w bazach danych. Lecz uwaga! Nie będą działały teraz programy w Basic'u zawierające procedury w kodzie maszynowym.

```
program: basic $b00  00801 0a8d
-------------------------------------
0801 : 1a 08 c2 07 9e 32 35 36 6e
0809 : 30 3a 12 20 42 41 53 49 ed
0811 : 43 20 24 42 30 30 30 20 3b
0819 : 00 00 00 a9 a0 85 44 a9 e9
0821 : b0 85 46 a9 e0 85 48 a0 f7
0829 : 00 84 43 84 45 84 47 b1 c6
0831 : 43 91 45 b1 47 91 47 c8 74
0839 : d0 f5 e6 44 e6 46 e6 48 13
0841 : d0 ed a9 35 85 01 a9 b0 81
0849 : 8d 01 fd 8d 8b fd 8d 71 4a
0851 : fe a0 0d a2 81 20 c6 08 7b
0859 : a9 b3 a0 29 a2 65 20 c6 8c
0861 : 08 a9 e4 a0 48 a2 54 20 b6
0869 : c6 08 a9 c8 8d d7 08 a9 c2
0871 : b0 a0 82 a2 a0 8e 8a fd 0b
0879 : 20 c6 08 a9 b3 a0 8a a2 e3
0881 : d0 20 f9 08 a9 e0 a0 00 05
0889 : a2 e5 8c 89 fd 8e d6 fd 1e
0891 : 20 f9 08 a9 a2 8d 88 fd 9a
0899 : a9 b3 a2 d0 a0 00 20 36 d5
08a1 : 09 a0 8a 20 3d 09 a9 e0 25
08a9 : a2 e3 a0 3d 20 36 09 a0 26
08b1 : 00 20 3d 09 a9 e3 a2 e4 40
08b9 : a0 5e 20 36 09 a0 7b 20 1b
08c1 : 3d 09 6c fc ff 85 44 8e 98
08c9 : d9 08 b1 43 c9 e0 b0 04 e9
08d1 : 69 10 91 43 c8 c8 ea c0 0f
08d9 : aa d0 ef 60 c8 d0 02 e6 dc
08e1 : 44 a5 46 c5 44 d0 08 c4 b7
08e9 : 45 d0 04 ba e8 e8 9a 60 f0
08f1 : c0 00 d0 02 c6 44 88 60 97
08f9 : 85 44 86 46 b1 43 c9 a9 ba
0901 : d0 2d 20 dd 08 20 dd 08 35
0909 : b1 43 c9 a0 d0 1b 20 dd 04
0911 : 08 b1 43 29 f0 c9 a0 f0 aa
0919 : 04 c9 b0 d0 09 b1 43 18 a3
0921 : 69 10 91 43 d0 09 20 f1 19
```

— LIGHT PEN —

C-64

Na naszych łamach przedstawialiśmy już kilka urządzeń współpracujących z C-64. W dzisiejszej publikacji zajmiemy się piórem świetlnym. Jest ono jednym z ciekawszych narzędzi służących do komunikowania się użytkownika z komputerem.

Sposób jego obsługi przypomina pisanie zwykłym piórem po arkuszu papieru, z tą różnicą, że papier zastępujemy ekranem monitora a pióro posiada elementy elektroniczne. Pióro takie można stosować do wykonywania prostych czynności jak: rysowanie na ekranie jednorodnych linii, okręgów, trójkątów a także do n.p. rysowania wielobarwnego tła, odwracania rysunków, powiększania ich itp. Bardziej skomplikowane czynności wybierane są z menu programu. Możliwości zależą tylko od stopnia rozbudowania programu - od zabawy do szeroko rozumianego projektowania.

Zasada pracy pióra jest prosta. Wykorzystuje się w niej sposób pracy monitora. Tutaj naszym czytelnikom należy się garść informacji na temat pracy monitora. Złożony sygnał wizyjny, bo taki powstaje w komputerze, poprzez układy elektroniczne steruje powstawaniem obrazu w monitorze. Obraz widoczny na ekranie składa się z drobnych punktów o zróżnicowanej jasności. W monitorze kolorowym poszczególne punkty mają dodatkowo zróżnicowane barwy. Wymienione punkty kreślone są przez odpowiednio sterowaną wiązkę elektronów, wysyłaną z elektrod kineskopu. W przypadku grafiki C-64 możliwe jest sterowanie 320 punktami w liniach poziomych (współrzędne x), a w pionie 200 (współrzędne y).

Wszystkie punkty na ekranie wyświetlane są z częstotliwością 50 Hz tzn. 50 razy w ciągu sekundy. Bezwładność ludzkiego oka powoduje wrażenie stałości obrazu. Element światłoczuły umieszczony na końcu pióra z łatwością odczytuje te zmiany i poprzez ekład elektroniczny zawarty w piórze emituje impuls. Na tej podstawie program obsługi pióra świetlnego szacuje jego położenie na ekranie.

Nasz poczciwy C-64 przystosowany jest do współpracy z piórem świetlnym - dpowiedzialny za nią jest procesor graficzny VIC (videocontroler). Do połączenia piora świetlnego z komputerem służy port nr 1. Wspomniany wcześniej impuls z pióra powoduje zapamiętanie w rejestrach 19 i 20 VIC-a aktualnych współrzędnych wiązki elektronów kreślącej obraz na ekranie:

wzdłuż osi x
VIC+19=53267 ($ d013)
wzdłuż osi y
VIC+20=53268 ($ d014)

Jak już wiemy, na ekranie graficznym mamy 320 możliwych pozycji na osi x, a do rejestru możemy wpisać max. wartość 255 ($ ff).

W programach wykorzystuje się wartości od 0 do 160 (określenie położenia z dokładnością do dwóch punktów). Aby otrzymać dokładną informację o położeniu pióra wartość odczytaną z rejestru 19 należy pomnożyć przez 2.

Problem ten nie występuje przy określaniu położenia w osi y, gdyż pozycji w tej płaszczyźnie mamy jedynie 200. Dlatego też liczby odczytane z rejestru 20 będą zawsze zawierały się w przedziale od 0 do 200.

A teraz wykonajmy następujący program w BASIC'u:

```
10 VIC = 53248
20 X = PEEK ( VIC + 19 )
30 Y = PEEK ( VIC + 20 )
40 PRINT X ; Y
50 GOTO 10
```

Liczby pojawiające się z lewej strony ekranu to zawartość rejestrów VIC-a. Przez zbliżanie pióra świetlnego do ekranu w różnych punktach powodujemy zmianę wartości tych liczb. Adaptując wartości z rejestrów do określania położenia pióra świetlnego na ekranie niskiej rozdzielczości otrzymamy:

K = (X - 35)/4
gdzie K - numer kolumny
X - zawartość rejestru X VIC-a
W = (Y - 50)/8
gdzie W - numer wiersza
Y - zawartość rejestru Y VIC-a

Wartości 35 i 50 są zależne od rodzaju monitora. Najczęściej dobiera się je doświadczalnie.

Te podstawowe wiadomości pozwalają na konstrukcję prostego programu obsługi pióra świetlnego. Programy bardziej rozbudowane, pisane w assemblerze 6510, również testują zawartość rejestrów VIC-a. Ale o tym innym razem.

Na zakończenie pragniemy podziękować firmie AUTOSOFTWARE, która udostępniła do testowania pióro świetlne. Firma ta przygotowała użytkownikom miły prezent - w najbliższym czasie będzie prowadzić sprzedaż piór świetlnych w cenach przystępnych dla przeciętnego nabywcy (dla zainteresowanych adres firmy poniżej).

UWAGA KONKURS!!!

AUTOSOFTWARE ogłasza konkurs na program obsługi pióra świetlnego w dwóch kategoriach :

- Program w BASIC
- Program w języku maszynowym

Najciekawsze, nie publikowane programy będą nagrodzone. Nagrodami są oczywiście pióra świetlne do C-64!

Prace prosimy nadsyłać w terminie do 30 listopada br. na adres:

AUTOSOFTWARE
85-338 BYDGOSZCZ
ul. Czerwonego Krzyża 39

C-64

43007 WOLNYCH BAJTÓW!

Poniższy program przesuwa interpreter Basic'a z $a000-$BFFF na $B000-$CFFF oraz zmienia wszystkie parametry, wartości i adresy.

Basic zamelduje się teraz ilością 43007 zamiast 38911 wolnych bajtów. Zaletą tej metody w stosunku do rozszerzeń RAM'ów jest to, że pamięć jest do dyspozycji w jednym obszarze.

Dodatkowe 4kB można użyć w grach typu adventure względnie w bazach danych. Lecz uwaga! Nie będą działały teraz programy w Basic'u zawierające procedury w kodzie maszynowym.

```
program: basic $b00 00801 0a8d
--------------------------------------
0801 : 1a 08 c2 07 9e 32 35 36   6e
0809 : 30 3a 12 20 42 41 53 49   ed
0811 : 43 20 24 42 30 30 30 20   3b
0819 : 00 00 00 a9 a0 85 44 a9   e9
0821 : b0 85 46 a9 e0 85 48 a0   f7
0829 : 00 84 43 84 45 84 47 b1   c6
0831 : 43 91 45 b1 47 91 47 c8   74
0839 : d0 f5 e6 44 e6 46 e6 48   13
0841 : d0 ed a9 35 85 01 a9 b0   81
0849 : 8d 01 fd 8d 8b fd 8d 71   4a
0851 : fe a0 0d a2 81 20 c6 08   7b
0859 : a9 b3 a0 29 a2 65 20 c6   8c
0861 : 08 a9 e4 a0 48 a2 54 20   b6
0869 : c6 08 a9 c8 8d d7 08 a9   c2
0871 : b0 a0 82 a2 a0 8e 8a fd   0b
0879 : 20 c6 08 a9 b3 a0 8a a2   e3
0881 : d0 20 f9 08 a9 e0 a0 00   05
0889 : a2 e5 8c 89 fd 8e d6 fd   1e
0891 : 20 f9 08 a9 a2 8d 88 fd   9a
0899 : a9 b3 a2 d0 a0 00 20 36   d5
08a1 : 09 a0 8a 20 3d 09 a9 e0   25
08a9 : a2 e3 a0 3d 20 36 09 a0   26
08b1 : 00 20 3d 09 a9 e3 a2 e4   40
08b9 : a0 5e 20 36 09 a0 7b 20   1b
08c1 : 3d 09 6c fc ff 85 44 8e   98
08c9 : d9 08 b1 43 c9 e0 b0 04   e9
08d1 : 69 10 91 43 c8 c8 ea c0   0f
08d9 : aa d0 ef 60 c8 d0 02 e6   dc
08e1 : 44 a5 46 c5 44 d0 08 c4   b7
08e9 : 45 d0 04 ba e8 e8 9a 60   f0
08f1 : c0 00 d0 02 c6 44 88 60   97
08f9 : 85 44 86 46 b1 43 c9 a9   ba
0901 : d0 2d 20 dd 08 20 dd 08   35
0909 : b1 43 c9 a0 d0 1b 20 dd   04
0911 : 08 b1 43 29 f0 c9 a0 f0   aa
0919 : 04 c9 b0 d0 09 b1 43 18   a3
0921 : 69 10 91 43 d0 09 20 f1   19
0929 : 08 20 f1 08 20 f1 08 20   b1
0931 : dd 08 4c fd 08 85 44 86   b0
0939 : 46 84 45 60 a2 4a b1 43   e8
0941 : dd 7d 09 f0 0f ca d0 f8   ba
0949 : a2 30 dd c7 09 f0 0e ca   59
0951 : d0 f8 f0 03 20 dd 08 20   8b
0959 : dd 08 4c 3d 09 20 dd 08   0e
0961 : 20 dd 08 b1 43 29 f0 c9   7d
0969 : 90 f0 08 c9 a0 f0 04 c9   e2
0971 : b0 d0 e4 b1 43 18 69 10   b3
0979 : 91 43 4c 58 09 69 29 c9   de
0981 : e0 c0 49 a9 a2 a0 09 e9   70
0989 : 65 25 06 24 c5 e4 c4 c6   ab
0991 : 45 e6 a5 a6 a4 46 05 26   64
0999 : 66 e5 85 86 84 75 35 16   19
09a1 : d5 d6 55 f6 b5 b4 56 15   9a
09a9 : 36 76 f5 95 94 b6 96 90   c5
09b1 : b0 f0 30 d0 10 50 70 71   28
09b9 : 31 d1 51 b1 11 f1 91 61   07
09c1 : 21 c1 41 a1 01 e1 81 6d   47
09c9 : 2d 0e 2c cd ec cc ec 4d   cd
09d1 : ee f9 20 ad ae ac 4e 0d   1d
09d9 : 2e 6e ed 8d 8e 8c 7d 3d   29
09e1 : 1e dd de 5d fe bd bc 5e   df
09e9 : 1d 3e 7e fd 9d 79 39 d9   c3
09f1 : 59 b9 be 19 4c 99 6c 00   3d
09f9 : 00 00 00 00 00 00 00 a9   4d
0a01 : 37 85 01 a2 00 bd 48 0a   b3
0a09 : 20 d2 ff e8 e0 29 d0 f5   36
0a11 : 20 9f ff a5 cb c9 27 f0   3f
0a19 : 19 c9 22 d0 f3 a9 b1 8d   28
0a21 : 36 08 a9 b0 8d 49 f0 8d   de
0a29 : 1e f4 a9 90 8d 4e f0 8d   68
0a31 : 23 f4 a2 00 bd 78 0a 20   7f
0a39 : d2 ff e8 e0 12 d0 f5 4c   79
0a41 : 1c 08 2a 2a 2a 2a 2a 93   f5
0a49 : 11 11 11 11 48 41 42 45   6b
0a51 : 4e 20 53 49 45 20 48 59   d7
0a59 : 50 52 41 2d 4c 4f 41 44   95
0a61 : 20 47 45 4c 41 44 45 4e   e8
0a69 : 20 3f 20 28 4a 2f 4e 29   e0
oa71 : 00 00 00 2a 2a 2a 2a 93   7a
0a79 : 11 11 11 11 20 42 49 54   5b
0a81 : 54 45 20 57 41 52 54 45   ed
0a89 : 4e 00 00 00 cc 00 02 c8   3e
```

Instrukcja obsługi programu:

Załadujcie nasz program i wystartujcie go RUN'em. Po odczekaniu około 7 sekund komputer zamelduje się 43007 wolnymi bajtami.

Wykonując SYS64738, komputer nie przejdzie do normalnego Basic'a lecz zostanie na nowo zainicjowany.

Możemy to wykonać klawiszem RESET lub instrukcją POKE 1,55. Aby znów wystartować nasz program należy go ponownie załadować i uruchomić. Niestety nie możemy używać w tym wypadku programu TURBO TAPE, gdyż wykorzystuje on obszar od $C000 do $CFFF. Najlepiej program nasz współpracuje ze stacją dysków.

opr. R.G.

Na to czekali wszyscy!

AMIGA
MAGAZYN

Już od września na polskim rynku nowe pismo poswięcone komputerom Commodore AMIGA! Po raz pierwszy polski magazyn komputerowy będzie zajmował się profesjonalnie zagadnieniami użytkowników AMIGA. Przy współpracy z niemieckim wydawnictwem Markt & Technik, na 80 kolorowych stronach, wiadomości o Twoim komputerze, na które czekałeś od dawna. Ciekawe programy, najświeższe informacje, zastosowania komputerów Amiga o których jeszcze nie wiesz! Cała wiedza o Twojej maszynie!
Czy jeszcze się zastanawiasz?
Przekonaj się sam - warto!
*Już od września Magazyn AMIGA w kioskach *całej Polski!*



Wydawnictwo LUPUS

Pokwitowanie dla Wpłacającego
zł 240.000-
słownie dwiescie czterdziesci tysięcy zł
wpłacający Jan Kowalski
ul. Świętokrzyska
00-001 Warszawa
adres
na rachunek: LUPUS Sp. z o. o.
Warszawa, ul. Stępińska 22/30
IX Oddział PKO BP w Warszawie
r-k. nr. 1599-318121-136
Opłata
zł
datownik podpis przyjm.

dostępny także w prenumeracie rocznej (patrz wzór kuponu).

DLA WSZYSTKICH

OGŁOSZENIA

CO PIĄTY PROGRAM GRATIS! Studio Komputerowe PROLAIN oferuje wysyłkowo oprogramowanie, cartridge, literaturę do komputerów: **COMMODORE 64/128, AMIGA, IBM.** Niskie ceny, bogata oferta. Katalogi gratis (koperta+znaczek). PROLAIN, ul. Wyżynna 20/8, 20-560 Lublin.

Sprzedam Commodore 128D, monitor Neptun 156B, myszkę, joysticki, cartridge'e, literaturę, oprogramowanie, stan b. dobry, (5.000.000,-), Jarosław Skłodowski, 11-015 Olsztynek, Świerczewskiego 3/2, tel. 19-28-55.

Usługi Komputerowe "Gamex". Firma wysyłkowa. Literatura i oprogramowanie, Spectrum, Commodore, Amiga, IBM. Informacja: koperta i znaczek, 90-418 Łódź, Al. Piłsudskiego 14, D.H. Juwentus

---

Zamienię na Amigę 500 z modulatorem TV - C64, stację 1541 II (gwarancja), OTV cz/b, mysz, joy., cardr.: X, Kernal, BB4, Final 3, Action Replay, ok 100 dyskietek + box, mouse pad, kasety, literatura, magnetofon. Jacek Stucki, ul. Główna 47, 89-500 Tuchola

---

THE GROUP AGNUS (działająca na C-64) nawiąże kontakt z uczącymi się Assemblera. Uwaga! Poszukujemy: Swaperów, Muzyków, Grafików, Lamerów, Coderów. Rafał Tylus, ul. Niepodległości 16/36, 62-400 Słupia

---

Sprzedam Amigę 500, 2.5 MB RAM (ze zmianą wielkości i typu pamięci), Boot Selector, modulator, 2 joystiki, stacja dysków 5.25", oraz literatura. Cena 9 mln. W całości lub pojedyńczo. Tadeusz Rzok, ul. Słowiańska 16/8, 41-503 Chorzów.

Amiga - najlepsze i najnowsze gry, użytki. Wysyłka pocztą, najniższe ceny, katalogi (koperta + znaczek), Jacek Pałka, ul. Brzozowa 79, 13-230 Lidzbark Welski.

Stację dysków na raty kupię. Lub zamienię C-64, magn. ok 15 kaset. + ZENIT E+, książki o tem. techn. (C-64 gwar. do 12.93 r.) na Amigę 500. Zbigniew Leśniewski, 19-203 Grajewo, oś. Południe 6/25

Kupię "64 plus 4 & Amiga" nr 6/92. Krzysztof Gibas ul. Krawiecka 5/2, 45-024 Opole

---

Komputerowa firma usługowa "TREND", Commodore, Amiga 500-3000, literatura w j. polskim (!) i oprogramowanie. Informacja: dyskietka lub koperta + znaczek. Kontakt: Rafał Wierzbicki, ul. Budziszyńska 112/28, 54-436 Wrocław, tel. 57-71-23

Zamienię C64II na Amigę 500 (używaną). Wiadomość koperta + znaczek lub tel. 56-67-10, Piotr Duszyński ul. Startowa 7/C m. 26, 80-461 Gdańsk-Zaspa

---

Wszystko do Amigi 500/2000 po najniższych cenach: gry, dema, użytki oraz super nowości, polskie wersje programów i instrukcje, peryferia MK III, stacje dysków, dyski 5.25", 3.5" DD, HD, Boot Selectory DF0-DF3. Adres: THE HAUNTED, ul. 1000-lecia 6/203, 41-303 Dąbrowa Gór., telefon: 0/32/644-544. Sprzedaż tylko wysyłkowa. Hurt.

Interesujących się Aztec C proszę o kontakt. Marcin Nosek, Goworowska 4/28, 07-400 Ostrołęka.

Uwaga! Bezpłatna pożyczka do 12 mln i więcej, koperta A5 + zn. Marcin Kalinowski, ul. Bałkańska 9 m 45, 85-167 Bydgoszcz.

Kupię 1,2,3,5 numer TOP SEKRET, Jarosław Wita, ul. Odrodzenia 48, 44-234 Leszczyny.

Sprzedam: nową Amigę 2000C (gwarancja), ActionReplay III (gwarancja), Monitor Commodore 1084S. Jan Grodecki, ul. Heleny 10/107, 30-838 Kraków.

Wymienię oprogramowanie na Amigę oraz C64 (taśma). Poszukuję programów do tworzenia grafiki wideo na Amigę. Kupię sampler do Amigi. Kontakt: Piotr Pacyna, ul. Poznańska 3, 64-200 Wolsztyn, Karpicko.

Sprzedam rozgałęziacz do dwóch magnetofonów do C-64 (50 tys.). Dec Roman, 36-140 Sokołów, ul. Podstawskiego 21.

Sprzedam pisma: MICROKLAN (86-87), IKS (86-89), KONKRET (86-88), MŁODY TECHNIK (82-91), literaturę o ZX-Spectrum. 02-764 Warszawa, ul. Iberyjska 4/2, tel. 42-62-61.

Sprzedam: osprzęt, literaturę, oprogramowanie do Commodore 64/128. Tomek Kępa, ul. Stawisińskiego 4f/68, tel. 488-523, 87-100 Toruń.

Magazyn PUBLIC 64, dysk/kaseta. Info. koperta + znaczek lub nośnik. Al. ul. Lompy 8/III/5, Pyskowice 44-120.

Sprzedam lub zamienię na Amigę 500 C-64, magnet., X, 2 joysticki, 1000 gier 3.0 mln (magnetofon i klawiatura na gwarancji). Damian Łoziński, ul. Szopy 3/40, 80-762 Gdańsk.

Sprzedam C-64II, magnetofon, joysticki, Black Box 4, 300 programów, cena 1,7 mln. Wojciech Chałat, ul. Aleja 8/44, 42-200 Częstochowa.

Sprzedam Amigę 500 2,3 MB + joysticki + modulator + 200 dyskietek + literatura. Grzegorz Karolczak, ul. Wielka 53/24, 53-338 Wrocław.

Sprzedam C+4, monitor, magnetofon, 250 pr. literaturę (2.300.000). M. Knap, ul. Cmentarna 3a, 58-377 Dobromierz.

Sprzedam A500 1MB + monitor mono + stacja 5.25" + modul. + 150 dysków + wiele dodatków - 11 mln zł. z LC20 14,5 mln. Piotr Mikulski, Adamieckiego 11/115, 41-300 Dąbrowa Górn.

Amiga 500/600/plus/3000 - najlepsze gry i programy użytkowe, nowości. Sprzedaż wysyłkowa pocztą. Ekspresowe terminy, katalogi gratis. "SOFTSTUDIO", Tysiąclecia 54/6, 31-610 Kraków, tel. (012) 48-51-50.

Sprzedam AMIGĘ 500, stację dysków, dyskietki. Józef Bała, Bydgoszcz, tel. 43-04-19.

Rewelacyjne programy użytkowe, oprogramowanie, opisy gier, 100% kopier na C-64. Inf. (koperta+znaczek) lub taśma. Skit, Strzelce Małe 17, 63-803 Bodzewo.

AMIGA "The home Computer Club"
Ul. I Armii Wojska Polskiego 4/41, 43-300 Bielsko-Biała

- Pomoc początkującym.
- Dyskietki.
- Bezpłatny dysk klubowy.
- Opisy - instrukcje.
- Wolny dostęp do klubowych programów.

Informacja = koperta zwrotna + znaczek!

COMMODORE 64
AMICOS COMPUTER
Ul. Wodzickiego 84/90/27, 42-200 Częstochowa, tel. 22-22-38

- ATRAKCYJNE PROGRAMY NA KASETACH
- CIEKAWE PROGRAMY NA DYSKACH
- KATALOGI GRATIS (koperta+znaczek za 2.5tys. zł)
- WYSYŁKA POCZTĄ EKSPRESOWA

PUBLIC DOMAIN PACK

PUBLIC DOMAIN PACK C-64

Styczeń '91 (nr 1)

- Mega demo grupy „VISION"-MIST2, Preview do gier: UN SQUADRON, PUZZLENOID, TURRICAN.

Luty '91 (nr 2)

- Tune of month, Logo Writer V 2.0, Fast Cruelcrunch, WRATH+ (DEMO)[02], Dreptacz - BASIC, SWISS CHEESE/CFA, Disk Fast Loader.

Marzec '91 (nr 3)

- Font Grub 1.0, Projektant Duszków, Strzałka 64+, Piratek - gra, V4.0 - Symphonies, Cruiser, The First, Commercial Break, Relakator 64, Korektor 64, Flash, HOT SHOT nr 9 (zach. mag.), BAD NEWS nr 2 - j.w., demo - rekord - 290 sprite'ów!, demo: NEW INTRO, demo: LET'S DYCP, Kontakt Corner - adresy, New Fast - działa z 1541 I 1541 II, CSLINKER V2.0.

Kwiecień '91 (nr 4)

- Digi - Organizer - program do tworzenia muzyki z użyciem digitalizacji dźwięku, „ONE YEAR - RADIUS" - mega demo grupy RADIUS.

Maj '91 (nr 5)

- CRUEL SOLIDERS - demo, DESTINATION - demo, SUCKER DJ! - demo (digi mix), MUSIC SEARCHER - do wycinania ilustracji muzycznych z programów, MEGA DEMO „INFOSYSTEM 91".

Czerwiec '91 (nr 6)

- Fonteditor, Sindata Editor, Color Editor, Disk - Noter, Gwiazdy - demo graficzne, FILGRAEPH 2.2/BML, NOTE TO FLI V 2.2, AFLI - EDITOR V 1.2, RESET - MON,8,1, TURBO - ASS 5, HIGHLIFE #5, AXEL NEWS #1, DISK NOTKA/PADUA, PSC - MAG #9'06/91, CONSPIRE? OREGON - demo, CONTACT DEMO/ORE, SHOWPIX.

Lipiec '91 (nr 7)

- Mega demo „MY, OH MY!" grupy LIGHT, Game Music Composer - edytor muzyczny grupy GRAFFITY z Węgier.

Sierpień '91 (nr 8)

- MegaDemo „Unnamed" grupy CAMELOT, Sound Killer - edytor muzyczny grupy TOPAZ, AFLI - edytor graficzny techniki A-FLI, Disk-Dos obsługa komend stacji dysków, Noter v2.2 grupy TOPAZ, IFFL - Squeezer kompresor dyskowy, Dismaster+ - edytor do dyskietek, Super Copy - DOS szybki program kopiujący do zbiorów, Mega Demo fińskiej grupy TOPAZ - „Graveyard Blues".

Wrzesień '91 (nr 9)

- Mega Demo grupy FLASH, Hot Shot - magazyn dyskowy, Code Sucker monitor - pr. użytkowy grupy PADUA, Mountain Ride - gra w BASIC.

Październik '91 (nr 10)

- MEGA DEMO „AIRDANCE 4" grupy T.A.T.

Listopad '91 (nr 11)

- NEW LAW & ORDER, FLT/LEGOLAND, FLT/LEGONOTE, TERMINAT .2%/FLT, SM. CRIMINAL #8, SMALL BUT FINE, HOLLY SMOKE/M12, UNITE!/SYLVIO.

Grudzień '91 (nr12)

- Armageddon 3, NOTE TO DEMO, OUTRUN 2 MUS $ SFX, AFTERBURNER/MON, TRIVIA-GAME MUSIC, FORM.I. SIMULATOR, 2400AD END-TUNE, NIGHTHUNTER MUSIC, TOMCAT MUS./MON, ZAMZARA TUNE/MON, NOTE TO DISK, HIGHLIFE #9.

Styczeń '92 (nr 13)

- Char Zoomer v3.1, Colour Bar Editor v3.0, Hires+A-FLI Editor, FLI Designer v1.1, Accesinus, Music Routine Cruncher v1.5, Gandalf Protector v3.0, Gandalf Coder v1.0, Disk-tape copy v.20, Gnd - packer v1.0.

Luty '92 (nr 14)

- LYNX XVI+, Sideborder Logo Editor V1.0, Intelpaint, Flieditor V3.2, 4*4 Charmaker, F(R)ONT EDITOR 3, THE GRAFIX PACK II, DEMA.

Marzec '92 (nr 15)

- Contact Dealer v3.0, Beeftrucker v1, Cross linker v3, PowerCruncer v7.1, Sample mon v.2.0, Handy Term v8.4, Highlife, Darkside, Humor Basic, dema.

Kwiecień '92 (nr 16)

- CIABACH 4&5, DARKSIDE, Convert Studio, edytor znaków grupy Skylight, Sprite Designer, Multicolour Conventer v1.3, Logo Flipper, Charset Maker v1.0.

PUBLIC DOMAIN PACK AMIGA

Styczeń '91 (nr 1)

- Programy kompresorów danych, Grafiki Borysa Vallejo, prezentacja najlepszych muzyczek, INTUITRACKER.

Luty '91 (nr 2)

- Request player; Multi ripper, 3-rd day; Phantasmagoria - demo, Master Seka; Virus Ekspert v1.6, AMOS-programy; Moduły: Killing game show, Upon Me, Let's swing it.

Marzec '91 (nr 3)

- PROTRACKER V1.0 (pakiet programowy). Najlepsze muzyczki: NOW WAIT? - DR.AWESOME, AMOS - procedury, DEMO grupy REBELES „TOTAL TRIPLE TROUBLE".

Kwiecień '91 (nr 4)

- RUBBER VECTORS - demo, KEFTALES - demo, DISK MASTER V3.0, Moduły muzyczne: TECHNOSTYLE 2, GALAXY 2, GRAFIKA - prezentujemy rysunki - RICK PARKS.

Maj '91 (nr 5)

- VIRUS X 5.0, VIRUS TERMINATOR, PARADOX - demo, STORMCHILD - demo, Moduły muzyczne: MIAMI VOICE, ANTI ATARI SONG.

Czerwiec '91 (nr 6)

- POWER BOOT - własne menu dysku, DISK CODING SYSTEM - program do zabezpieczania dysków, Konwerter IFF - ANSI, AUER NATION - demo, Moduły muzyczne, DOCS - opis gry ELWIRA, LAMER DEFENCE - do wykrywania i niszczenia wirusów, REWENG GO OF THE LAMER - grafika w trybie D-HAM.

Lipiec '91 (nr 7)

- Sanity - demo, Amiga - Tanx (1Mb) - gra, Little Beau (1MB) - gra, There is A Light/Tonid - modules.

Sierpień '91 (nr 8)

- Real 3D - demo programu do raytraceing'u, moduł muzyczny XTC STEREO.

Wrzesień '91 (nr 9)

- Moduły muzyczne dla programu TFMX: R - TYPE; The House of Techno;VIRUS EXPERT v181 + 143; Boot Block'i, Bootx v 3.80, Imploder v 4.0.

Październik '91 (nr 10)

- ANARCHY - „THE INSPIRATION IS NONE"; ‚DUAL CREW - „NEW DIMENSION"; SANITY - „ELYSIUM".

Listopad '91 (nr 11)

- COMPUTER HEAD - animacja; CONFUSED - moduły pod medplayer i wiele sampli; ROCKED -; SAVE GAME „MONKEY ISLAND".

Grudzień '91 (nr 12)

- GEM X; BOOTX V4.13; FINAL KIT - monitor; MEGA-MON; VARIA.

Styczeń '92 (nr13)

- Super Duper 2.01; Sanity Copy; Noise Packer 3.00; Ham Sharp; Mostra; dema graf. i muz.

Luty '92 (nr 14)

- Nuke Saddam 1.4; THIEF RIPPER 2.0; DISK MASTER 3.05; ZIG ZAG #3 (grafika); dema graf. i muz.

Marzec '92 (nr 15)

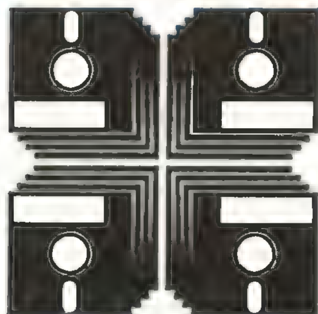
- Gry: KIM, Power Wars; PrtDrvGen; FileMaster v1.1; FixDisk!; Facc; Intro Welll; digitalizacje (IFF).

Kwiecień '92 (nr 16)

- Disk Cruncher; IBM-ST-AMIGA; HamLab v0.91; BAD v4.13; BOOTX v4.47,; DragonTiles - gra; grafiki IFF.

PUBLIC DOMAIN PACK





PUBLIC DOMAIN PACK C-64
TAPE NR 1

- TURBO
- SINUSDATA - EDITOR
- FAST CRUNCHER V3
- ANAL S.C. IBEYOND
- VECTOR - VICTORY
- PUZZLENOID+4
- TUNE OF MONTH #1
- NIM
- STRZAŁKA 64+
- LOGO - WRITER V.2.0
- CAN'T TOUCH IKU!
- NTRO PRV
- BONZIEED!!
- ZAX PACKIS
- READ THIS FIRST
- COMMERCIAL BREAK
- 290 SPRITES!
- NOTE - ABOUT
- BAD NEWS NR2
- TO BAD NEWS...
- CONTACT CORNER!
- PROJEKT DUSZKÓW
- SYMPHONY NR14
- SYMPHONY NR15
- SYMPHONY NR16
- SYMPHONY NR17
- SYMPHONY NR18
- SYMPHONY NR19
- CRUISER/GIANTS
- NOTE>ANO<PADUA
- LET'S DYSP!
- FINALTAPE
- MUSIC - SEARCHER

PUBLIC DOMAIN PACK C-64
TAPE NR 2

- TURBO
- PUBL. DOMAIN. INFO
- FONTGRUB 1.0
- DREPTACZ BASIC
- LOAD DIS FIRSY
- MACROASSEMBLER
- TURBOASSEMBLER
- RELOCATOR
- LOGOPAINTER 3!
- REASSEBLER
- SPRITE - EDITOR
- FAST - CRUEL U.2.5
- HIGHLIFE NR5
- AXEL NEWS NR1
- GWIAZDY
- FLIGRAPH 2.2/BML
- NOTE TO FLI V.2.2
- DISKNOTKA/PADUA
- MEGA PACKER/T
- MIST II/ VISION
- TTECHSCR &DYSP
- PLASMA - WORLD
- VECTORBOBS...
- VECTOR - PLOTS
- FLI - UPSCROLL
- BORDER - HIRES
- ROCK AROUND
- FACEWRITER
- CHAR EDIT 2+2
- DISKNOTER
- DESTINATION'91
- CONTACTDEMO/ORE
- FONTEDITOR
- THE END

PUBLIC DOMAIN PACK C-64
TAPE NR 3

- TURBO
- PUBLIC DOMAIN NOTE
- GRAVEYARD NOTES!
- NOTE FROM BEAT!!
- ANONYM SPEAKING!
- SNDK. V3.7/TOPAZ
- AFLI - EDITOR
- NOTER V2.2/TOPAZ
- DLW V1.5/TOPAZ
- CODE - S.MON/PADUA
- OPINION - POLL/PDA
- MOUNTAIN RAID
- PART 1
- PART 2
- PART 3
- PART 4
- PART 5
- FAIRLIGHT 1
- FAIRLIGHT 2
- FAIRLIGHT 3
- FAIRLIGHT 4
- FAIRLIGHT 5
- THE END

PUBLIC DOMAIN PACK C-64
TAPE NR 4

- TURBO
- OUT RUN 2 MUS & SFX
- AFTER BURNER/MON
- FORM.1.SIMULATOR
- 2400 AD.END - TUNE
- NIGHT HUNTER DIGI
- ELEMINATOR MUSIC
- TOMCAT MUSIX/MON
- ZAMZARA TUNE
- DYNAMIX TUNE
- HIGHLIFE NR9
- SNAKES C3
- SNARK C3
- SNERD C3
- WAREHOUSE C3
- STARTREK C3
- TOWER
- SNOOPY
- NEW LAW & ORDER
- FLT/LEGONOTE...
- TERMINAT.2%/FLT
- UNITE!/SYLVIO
- BALL - SCOPE/451
- TRIVIA - GAME MUS.
- RESET - MONITOR
- HOLY SMOKE

Zestawy „64 plus 4 PUBLIC DOMAIN PACK" można zamawiać wpłacając na konto: Bank PKO SA Oddział w Bydgoszczy konto nr: 5.09011-400522.7-2511-30-111.0 następujące kwoty: 20.000zł za pojedyńczy zestaw dyskowy dla C-64, 30.000 zł za zestaw programów PD na kasecie, 25.000zł za zestaw dla Amigi.

Blankiety wpłat powinny być CZYTELNIE wypełnione i zawierać: **imię i nazwisko, dokładny adres zamawiającego, skrót „PDP-64D"** - jeśli zamawiamy zestaw dla C-64 na dyskietce lub **„PDP-64T"** - dla zestawu taśmowego, zestaw dla Amigi prosimy zaznaczać skrótem **„PDP-A"** - dane te prosimy umieszczać na **wszystkich** odcinkach dowodu wpłaty.

W prenumeracie zestawy kosztują: **PDP-64 - 18.000zł** (12 numerów 216 tys. zł), **PDP-A - 22.000 zł** (12 numerów 264 tys. zł). Prenumeratę można zawrzeć w dowolnym terminie na okres od 3 do 12 miesięcy (do końca roku kalendarzowego). Powyższe warunki odnoszą się również do naszych zestawów wydanych w 1991r .

Zestawy taśmowe PDP-64 w 1992r. będą ukazywały się w miarę napływu nowych, ciekawych programów - o czym będziemy informować na łamach naszego pisma.

Zamów nie zwlekaj!

VOICETRACKER V4.0

C-64

Rewelacyjny program muzyczny!



Tylko **50.000 zł** kosztuje fantastyczny edytor muzyczny wykorzystujący ogromne możliwości dźwiękowe komputera Commodore - 64. Oferowany zestaw zawiera dyskietkę lub taśmę magnetofonową z programem VOICETRACKER V4.0, instrukcję obsługi, oraz - dodatkowo - przykładowe demonstracje muzyczne. **UWAGA! Wersja magnetofonowa tylko 40.000 zł!**

Przedsiębiorstwo ABUK posiada wyłączność na dystrybucję tego programu. Wszelkie kopiowanie programu i powielanie instrukcji jest zabronione. Nabywcy otrzymują rejestrowane kopie programu wraz z prawem nabywania nowych wersji po znacznie obniżonych cenach oraz wymiany dyskietki w razie uszkodzenia. Studiom komputerowym proponujemy zakup hurtowy (przy zakupie powyżej 10 kompletów udzielamy 20% rabatu).
Chcąc stać się posiadaczem programu VOICETRACKER V4.0 wystarczy dokonać wpłaty 50.000 zł (wersja dyskowa) lub 40.000 zł (taśma) na konto: Bank PKO SA Bydgoszcz, konto nr: 5.09011-400522.7-2511-30-111.0. Na blankiecie prosimy czytelnie podać swoje imię, nazwisko i adres wraz z dopiskiem „VV4.0" uzupełnionym literką „T" - taśma lub „D" - dyskietka.

W związku z pojawiającymi się kłopotami w dystrybucji oferowanych przez nas dyskietek i taśm (wynikającymi z nieczytelnego bądź niekompletnego wypełnienia blankietów wpłat) przedstawiamy obok specjalny druk. Blankiet ten może służyć jako zamówienie i dowód wpłaty dla wszystkich oferowanych przez nas usług: sprzdaż dyskietek i taśm PDP, Voicetracker'a, zamówienie ogłoszeń itd.

REDAKCJA

Odcinek dla wpłacającego

Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)

na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0.

Opłata
zł..........

Odcinek dla posiadacza rachunku

Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)

na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0.

Opłata
zł..........

Odcinek dla Poczty

Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)

na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0

Opłata
zł..........



TREŚĆ ZAMÓWIENIA:

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA:

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA:

Zapraszamy wszystkich do udziału w stałym konkursie pod hasłem:

Najlepszy program miesiąca

W konkursie udział mogą brać wszyscy, którzy nadeślą własne, nigdzie nie publikowane prace. Tematyka programów dowolna. Konkurs rozgrywany jest osobno dla komputerów C-64 i Amiga. Teksty programów należy nadsyłać na adres redakcji na dyskietce lub w postaci czytelnego rękopisu (dyskietki będą przez redakcję zwracane). Objętość programu wraz z opisem i komentarzem nie powinna przekraczać 4 stron maszynopisu. Raz w miesiącu Sąd Konkursowy wybierze najlepsze programy przyznając ich autorom dwie główne nagrody po

500.000 zł każda.

Decyzje Sądu Konkursowego są nieodwołalne. Oprócz zdobycia głównej nagrody autorzy mają szansę na publikację swych prac na łamach naszego pisma. Pracę prosimy podpisywać imieniem i nazwiskiem oraz dokładnym adresem autora.

Redakcja

WSZYSTKICH ZAINTERESOWANYCH NABYCIEM ZALEGŁYCH NUMERÓW

„64 plus 4 & AMIGA"

INFORMUJEMY, ŻE POSIADAMY JESZCZE OGRANICZONĄ ILOŚĆ NUMERÓW OD LISTOPADA 199OR. DO GRUDNIA 1991R.

ZAMÓWIENIA PROSIMY KIEROWAĆ NA ADRES :
Przedsiębiorstwo ABUK sp. z o.o.,
87-200 Wąbrzeźno,
ul. 1 Maja 33.

(Pod tym adresem mieści się dział kolportażu - tam też prosimy przesyłać wszelką korespondencję dotycząca kolportażu czasopisma, dyskietek, taśm itd. Adres redakcji się nie zmienił - patrz stopka.)

OPIS ZESTAWU PUBLIC DOMAIN PACK

nr 20 (sierpień '92)

No i koniec wakacji, ale mimo wszystko nasz dział Public Domain nie przerwał swojej działalności nawet na chwilę. Dla zapominalskich oraz nowych użytkowników Public Domain Pack'a opis funkcji Boot-block'a - na następnej stronie.

Na początek przyjrzyjmy się dokładnie jedynej grze z tego pack'a. Jest to karciana gra, albo będąc dokładnym - kilka gier pod banalnym tytułem **CardGames**. Niestety nie są to gry w rodzaju brydża, kierek, tysiąca czy podobnych (a szkoda!). Jest to zbiór zabaw przeznaczonych dla jednego gracza, wiekszość z nich polega na przekładaniu kart w tą i z powrotem, według jakichś "magicznych" zasad. W zestawie jest sześć gier o dość nieznanych i tajemniczych tytułach. Są to: **Pyramid, Golf, Klondike, Corners, Calculation** oraz **Canfield**. Można także zagrać w Grand Tour, czyli wszystkie gry po kolei. Gramy używając tylko myszki. Maksymalna liczba punktów za jedną rozgrywkę to 52 (jeden punkt za każdą kartę) plus bonus 50% co daje jakieś 26 punktów, a razem - 78 punktów, maksimum, które można uzyskać za cały turniej to 468 punktów. Opiszę tutaj pierwszą z tych gier, reszty musicie się dowiedzieć sami, gdyż na opisanie wszystkich nie starczy miejsca.

Pierwsza z gier to Pyramid. Jej celem jest usunięcie wszyskich kart piramidy na jedną kupkę. W piramidzie o podstawie złożonej z siedmiu jest dwadzieścia osiem kart. Tylko karty nie przykryte są dostępne w danej fazie gry. Karty są odkładane na kupkę automatycznie, w parach, których wartość wynosi razem 13 punktów. Oto punktacja kolejnych figur: As - 1 punkt, Walet - 11 punktów, Dama - 12 punktów, Król - 13 punktów. Reszta ma taką wartość jak nominalna na karcie tzn. dziesiątka - 10, dwójka -2 itd. Kolor nie ma znaczenia i wpływu na wartość karty. Król jest wyjątkiem i może (musi?) być zabrany pojedyńczo. Aby wybrać parę należy po prostu dwa razy pod rząd kliknąć na wybrane karty, raz na jedną i raz na drugą. Aby unieważnić wybór należy kliknąć na nią ponownie. Zgodnie z punktacją As może być usunięty z Damą, Walet z dwójką, dziesiątka z trójką itd. Zasada, dostępności karty gdy jest ona odkryta nie działa tylko w jednym przypadku - mianowicie wtedy, gdy karta jest przykryta przez swoją parę. W tej sytuacji można wybrać najpierw kartę odsłoniętą, a następnie tę drugą. Gdy nie ma już możliwości usunięcia kart ze stołu i nie można znaleźć już żadnej pary - należy użyć kart ze stosiku w lewym górnym rogu. Jeżeli nawet taka karta nam nie pasuje - należy przełożyć ją na kupkę obok. Aby wygrać grę wszystkie karty muszą znaleźć się na kupce głównej.

Należy zauważyć, że grafika w grze jest całkiem elegancka, a gra (o ile zna się zasady!) wciągająca. Szczególnie efektownie wyglądają figury - uważam je za całkiem udane, na niezłym poziomie.

Ok. odrywamy się od karcianego szaleństwa!

Użytek o nazwie **LastHope**, cóż to takiego? LastHope to program, który dokonuje odzysku programów skasowanych nieopatrznie z naszego dysku. Wiele jest programów, które to robią np. FixDisk, DiskMonPro, QuaterBackTools i inne ale tylko LastHope może uratować plik, którego nagłówek jest zniszczony. Właśnie dlatego został nazwany tak a nie inaczej (LastHope czyli Ostatnia Nadzieja) - jeżeli zawiedzie ten program to nie ma już sensu próbować innych, plik jest nie do odzyskania. Program zasługuje na pochwałę nie tylko za swoje unikalne możliwości, ale także ze względu na bardzo estetyczną oprawę graficzną. Niestety LastHope działa tylko ze starym systemem zapisu danych (Old File System), Fast File System jest mu nieznany. Myślę, że po stracie bardzo ważnego pliku zanim zaczniesz sobie wyrywać włosy z głowy i niszczyć ze wściekłości najbliższe otoczenie (krzesła, stoły, DYSKI itd.) warto spróbować najpierw użyć ten program. I jeszcze jedna rada - NIGDY nie używajcie DiskDoctora, z dysku workbencha. Program ten ma jedną wielką wadę - natychmiast zapisuje odnalezione pliki na tym samym dysku, na którym je znalazł. Co to oznacza nie muszę chyba wyjaśniać. A więc zawsze zanim spróbujesz użyć LastHope, spróbuj FixDisk'a lub DiskMonPro, nigdy Diskdoctora!

AMIGA

Kolejnym użytkiem, który znalazł się na PDP-Amiga w tym miesiącu jest **GfxConverter**. Już słysząc nazwę wiadomo co to za programik. Jest to oczywiście konwerter do grafiki. Program ten pozwala na swobodne operacje na grafice. Jego głównym przeznaczniem jest konwertowanie grafiki zapisanej w formacie IFF na plik, który zawiera same dane binarne. Jest to przeznaczone głównie dla koderów, do wykorzystania w demkach i grach pisanych w języku maszynowym. Zawiera on bardzo przydatną opcję zapisywania kolorów w formacie gotowym do użycia w copperliście. Bez tego byłoby to bardzo trudne. Nie dość tego, program pozwala na zapisanie kolorów na dwa sposoby. Pierwszym z nich jest Labels. Copperlista stworzona przez niego wygląda tak:

```
DC.W COLOR00,$0000
DC.W COLOR01,$0FFF
DC.W .....
```

Natomiast tak wygląda copperlista (a dokładnie - jej część) stworzona dzięki Registers:

```
DC.W $0180,$0000
DC.W $0182,$0FFF
DC.W .....
```

Różnica jest minimalna. Przy użyciu pierwszego sposobu należy tylko pamiętać o zadeklarowaniu wartości stałych. Należy to zrobić mniej więcej tak:

```
color00 = $180
color01 = $182
color02 ....
```

W przeciwnym przypadku assembler po prostu nie będzie znał wartości stałych colorN. Oczywiście każdy koder doskonale o tym wie, ta uwaga była tylko dla początkujących.

Program GfxConverter może także zapisywać dane jako bobsy (z maską bądź bez), sprite'y (3 i 15 kolorowe). Posiada także wygodną opcję Sleep, ale uwaga - użycie tej opcji powoduje utratę obrazka, który znajduje się aktualnie w buforze.

Na koniec dużo rzeczy, dzięki którym rzednie mina posiadaczy ST. Obrazki, animacje, moduły... a wszystkie najwyższej klasy, przebrane z wielu - specjalnie dla Was.



AMIGA

Public Domain Pack

nr 21 (wrzesień '92)

Serdecznie witam w kolejnym, wrześniowym Public Domain Pack'u. Jego zawartość jest bardzo ciekawa, ale zanim rozpocznę jej przedstawianie chciałbym przypomnieć sposób obsługi boot-blocka. Boot-block pozostał niezmieniony ze względów praktycznych, a także estetycznych (bardzo przyjemnie jest się przyzwyczaić do jedego boot'a, którego znamy na wylot - znacznie gorzej jest gdy na każdym dysku znajduje się inny). Tak więc postanowiłem pozostawić boot-block autorstwa Orcrista z LSD, notabene jeden z najlepiej napisanych boot'ów.

Oto jego menu:

F1 - Memory On/Off - włączanie bądź wyłączanie fast ram'u.

F2 - Drives On/Off - włączanie lub wyłączanie zewnętrznych stacji dysków.

F3 - Hard reset - zimny start systemu, łacznie z ustawieniem na nowo wszystkich wektorów systemowych (zabija virusy w pamięci).

F4 - Install - nagranie tego boot-block'a na dysk w stacji DF0.

F5 - 60 Hz (ECS) - zmiana częstotliwości wyświetlania obrazu.

No dobrze, klikamy, czekamy kilka sekund i przystępujemy do zbadania zawartości dysku. Od razu widać, że nastąpiła kolejna zmiana ikon - jak myślę znów na lepsze. Zrezygnowałem z ikon "trójwymiarowych" ponieważ wyglądają źle pod systemem os 2.0. Ikony są za to lepiej wykonane i ciekawsze od poprzednich, które były mojego autorstwa.

Wrześniowy Public Domain Pack jest całkiem na poważnie - żadnych gierek, same użytki "okraszone" kilkoma iff'ami i dwoma muzyczkami. Nawet wspomniane grafiki to nie żadne digitalizacje czy rysunki, ale wspaniałe zbiory Mandelbrota. Naprawdę wyglądają niesamowicie!

Wspomniane rysunki otrzymano używając programu **MandelMoutains**. Program pozwala na uzyskania zbioru Mandelbrota. Na zbiory te zwrócił uwagę kilkanaście lat temu matematyk pracujący w firmie IBM. Najciekawszą cechą tych zbiorów jest samopodobieństwo do siebie, co jak wiadomo znamionuje również fraktale.

Ale wróćmy do programu. W programie MandelMountains oprócz możliwości ustawienia podstawowych parametrów zbioru jest także wiele ciekawych opcji, niektóre z nich to "bajery", ale większość jest przydatna i urozmaica eksperymentowanie.



Program posiada możliwość wyboru priorytetu procesu, który używa (dostępne są trzy wartośći 0, -1 i -5 przy czym 0 jest ustawione po uruchomieniu programu). Można też zmienić zasięg używanych kolorów, ale niestety same kolory są stałe i są to po kolei: czarny, niebieski, czerwony, karmazynowy, zielony, siny, żółty i biały. Standardowo sam zbiór mandelbrota jest reprezentowany przez kolor niebieski, ale można to dowolnie zmienić. Można także, co ciekawe zmienić precyzję obliczeń na podwójną. Spowoduje to uzyskanie bardziej realistycznego obrazu.

Kolejnym bardzo ciekawym programem jest - **SurfPlot**, który od razu wpada w oko ze swoją wspaniałą, wielką ikoną. Sam program służy do rysowania powierzchni w przestrzennym układzie współrzędnych przy zadanych funkcjach określających każdą ze współrzędnych. Program jak na niekomercyjny produkt jest napisany naprawdę świetnie, posiada bardzo estetyczną szatę graficzną i robi dobre wrażenie. Autor określa go jako Colaware co jest wymyślonym przez niego rodzajem oprogramowania. Pomysł polega na tym, że możesz wysłać autorowi co tylko chcesz jako wynagrodzenie (np. ciastko, puszkę coli, dysk itp.). Zachęcam do eksperymentowania z funkcjami, a szczególnie polecam obejrzenie wszystkich obiektów z katalogu Examples - są one bardzo ciekawe.

Następny program - **GVPinfo** to użytek podobny do SysInfo, "włącz mnie a powiem Ci jaka jest twoja Amiga" powinien głosić podtytuł tego programu, podaje on bowiem rzetelne informacje bez żadnych głupich komentarzy w rodzaju R.I.P czy Yawn! (tak jak w sysinfo). Program można również używać jako tester wnętrza Amigi (np. podczas zakupu komputera) gdyż przy uruchomieniu sprawadzane są wszystkie układy specjalizowane, procesor, ROM, itd.

Kolejny program - to gratka dla byłych koderów na spectrumnie - **symulacja** programowa procesora **Zilog Z80**! To naprawdę ciekawa propozycja, ale tylko dla części z Was. Ja szczerze mówiąc nie szaleję za tym programem z tego powodu, że moim pierwszym komputerem był stary, dobry C64. Warto tylko dodać, że program posiada pełne środowisko pozwalające na programowanie w asemblerze tego procesora np. disassembler, assembler, debugger itd. W celu uzyskania dokładniejszych wiadomości radzę przejrzeć plik Z80intro. Jako ciekawostkę chciałbym dodać, że program Z80 został napisany, aby sparawdzić na ile wiernie można zasymulować programowo dowolny procesor na innym - zupełnie różnym.

Ostatni program to najnowszy grajek do modułów spakowanych PowerPacker'em. Czego ten program nie potrafi zagrać! Wszystkie rodzaje modułów z Protrackerów, SoundTrackerów, Noisetrackerów, moduły z Oktalizera, Future Composera i innych, których już nie pamiętam.

No cóż, i w ten sposób dotarliśmy do końca Public Domain Pack'a nr. 21. Na pocieszenie zostaje jeszcze trochę obrazków i modułów, które wcisnąłem niemal na siłę do PDP-Amiga - wrzesień.

Bawcie się dobrze!

Michał "Amber" Gosztyła

KĄCIK POCZĄTKUJĄCEGO KODERA CZ. 16

Menus

AMIGA

Każdy, kto używa Amigi musiał się spotkać z menu, choćby używając WorkBencha. Wywoływalne są one poprzez wciśnięcie prawego przycisku myszy (wtedy pojawia się na górze ekranu spis dostępnych menu) a następnie dokonania wyboru poprzez wjechanie kursorem myszki na odpowiednią pozycję. Menu są bardzo proste do osiągnięcia, a także bardzo prosto je się tworzy. W dzisiejszym Kąciku Początkującego Kodera spróbujemy przybliżyć naszym czytelnikom, w jaki sposób tworzyć własne menu, co oferuje nam biblioteka intuition i w jaki sposób zachodzi komunikacja pomiędzy programem użytkownika a menu.

TWORZENIE MENU

Menu możemy dołączyć do każdego okna i gdy użytkownik wciska prawy przycisk myszki to menu pochodzące z aktywnego okna zostanie wyświetlone na górze ekranu.

Nasze przykładowe menu będzie wyglądało następująco:

```
System Programy Dane
```

Użytkownik po wciśnięciu prawego przycisku myszki może przesunąć kursor na jedną z pozycji menu i wtedy pojawi się "lista menu":

```
SYSTEM Programy Dane
Ładuj
Nagraj
Drukuj
Wyjdź
```

Użytkownik może teraz dokonać wyboru dowolnej pozycji poprzez wjechanie kursorem myszki na tę pozycję i puszczenie prawego przycisku myszki. Użytkownik może wybrać kilka menu poprzez wciskanie lewego przycisku myszki (podczas gdy prawy przycisk jest stale wciśnięty) na poszczególnych pozycjach menu, lub jeżeli chce wybrać kilka następujących po sobie pozycji może wcisnąć obydwa przyciski myszki i wykonać rych kursorem po tych pozycjach, które chce uaktywnić.

Każda pozycja menu może mieć jeszcze swoje subpozycje rozwijane jako listy menu (sub-lista menu). Na przykład:

```
SYSTEM Programy Dane
Ładuj
Nagraj
DRUKUJ  na drkarkę
Wyjdź   do zbioru
```

Poszczególne pozycje menu można osiągnąć także z klawiatury jednak dostępne są tylko te, które zdefiniował programista. Wywołanie takiego menu dokonuje się poprzez wciśnięcie prawego klawisza 'AMIGA' z odpowiadającym danej pozycji menu klawiszowi.

W naszym przykładzie programista może stworzyć możliwość osiągnięcia z klawiatury funkcji Ładuj poprzez wciśnięcie klawisza 'AMIGA' z klawiszem 'L'. Biblioteka Intuition automatycznie w menu umieści znak 'A' i dany klawisz przy nazwie danej pozycji menu.

Program nie odczuje żadnej różnicy czy użytkownik wybrał menu czy wcisnął odpowiadającą mu kombinację klawiszy.

Należy pamiętać aby pozostawić odpowiednią przestrzeń po prawej stronie danej pozycji menu aby zmieścił się tam znak 'A' i odpowiednia litera odpowiadająca klawiszowi. Jeżeli mamy na przykład szerokość listy menu wynoszącą 100 to musimy ustalić ją na 100+COMMWIDTH i wtedy wszystko będzie pasowało (dla ekranów uruchomionych w niskiej rozdzielczości podajemy wartość LOWCOMMWIDTH zamiast wartości COMMWIDTH).

POZYCJE MENU

W definiowanym przez nas menu mogą istnieć dwa typy pozycji menu:

- pozycje określające akcję (uruchamianie odpowiednich procedur, na przykład Ładowanie zbioru),
- pozycje określające parametry (ustawianie odpowiednich znaczników, na przykład czy program ma ładować zbiory spakowane PowerPackerem czy też ma je pomijać).

Pozycje określające akcję mogą być wybierane cały czas i za każdym razem gdy zostaną wybrane nasz program otrzyma odpowiedni komunikat. Natomiast pozycje określające parametry mogą być wybrane tylko raz (wtedy są uaktywniane) a za następnym razem są deaktywowane, następnie aktywowane i tak w koło... Istnieje możliwość deaktywowania innych pozycji po wybraniu jednej.

Jeżeli pozycja określająca parametry jest uaktywniona przed jej nazwą zostaje umieszczony mały znaczek weryfikacji "v". Jeżeli chcemy użyć własnego znaczka weryfikacji to przed uruchomieniem menu umieszczamy w strukturze NewWindow wskaźnik do struktury Image (NewWindow-CheckMark) zawierającej kształt graficzny pożądanego przez nas znaku.

Jeżeli używamy pozycji określającej parametr to przy ustalaniu szerokości listy menu musimy pozostawić po lewej stronie odpowiednią ilość miejsca na znaczek weryfikacji. Jeżeli używamy standardowego, dostarczanego przez intuition znaczka weryfikacji to do szerokości listy możemy dodać wartość CHECKWIDTH (odpowiadającą szerokości znaczka dostarczanego przez bibliotekę intuition). Dla okien otwartych na ekranie niskiej rozdzielczości wartość CHECKWIDTH zastępujemy wartością LOWCHECKWIDTH.

WZAJEMNE WYŁĄCZANIE (Mutual Exclude)

Ta specjalna i bardzo użyteczna funkcja działa tylko dla pozycji określających parametry. Sam pomysł polega na tym, iż w momencie gdy użytkownik wybierze daną pozycję określającą parametry i zostanie ona uaktywniona inne pozycje z nią połączone zostaną automatycznie zdeaktywowane. Dla przykładu w prostym edytorze tekstu, w którym mamy możliwość druku czcionką wytłuszczoną, pochyloną lub podkreśloną nie możemy drukować jednocześnie kilkoma z nich, czyli w momencie wyboru jednego ze stylów inne muszą zostać zdeaktywowane. W momencie gdy użytkownik wybiera na przykład Wytłuszczenie pozostałe style zostają zdeaktywowane. Wzajemne wyłączanie dział w ten sposób:

Jeżeli znacznik CHECKIT (znacznik ten oznacza pozycję określającą parametry) nie został ustawiony w tym menu (pozycja określająca akcję) to wzajemne wyłączanie nie będzie miało wpływu na tę pozycję.

Jeżeli znacznik CHECKIT został ustawiony (pozycja określająca parametry) i znacznik CHECKED jest ustawiony to biblioteka intuition automatycznie wyłączy go i pozycja zostyanie zdeaktywowana.

Każda pozycja w liście menu posiada swoje własne pole wzajemnego wyłączania (MutualExclude), w którym możemy określić, które pozycje na liście, w której znajduje się to menu mają zostać zdeaktywowane gdy ta pozycja zostanie uaktywniona. Bit zerowy w tym polu odnosi się do pozycji pierwszej, pierwszy do drugiej i tak dalej...

W naszy przykładzie wartości dla wzajemnego wykluczania ustawiamy następująco:

- dla Wytłuszczenia ustawiamy wartość MutualExclude na $00000006 (binarnie%00000000000000000000000000000110 - wyłączone mają zostać pozycje menu: numer 2 i numer 3, pozycja numer 1 pozostaje niezmieniona).
- dla Pochylenia wartość ta będzie wynosiła $00000005 (wyłączenie pozycji numer 1 i numer 3).
- dla Podkreślenia wartość ta będzie wynosiła $00000003 (wyłączenie pozycji numer 1 i numer2).

OTWIERANIE MENU

Jeżeli chcemy przyłączyć menu do danego okna to musimy wykonać następujące operacje:

1. Najpierw należy zdeklarować i zainicjować jedną lub więcej struktur Menu (to te pojawiające się na górnym pasku ekranu). Struktury Menu powinny być połączone razem poprzez odwołania jednej struktury do drugiej. Każda struktura Menu odpowiada za jedną listę pozycji menu i każda ma przypisaną taką listę (a właściwie pierwszą pozycję w tej liście).

2. Następnie deklarujemy i inicjujemy struktury MenuItem odpowiadające za każdą pozycję w liście menu i po połączeniu ich w całość podłączamy je do struktury Menu. Jedna struktura MenuItem reprezentuje jedną pozycję w liście menu. Do każdej pozycji w menu możemy dołączyć subpozycje, które są takimi listami menu jak pozycje normalne (sub-pozycje używają nawet tych samych struktur (MenuItem) jak i innych danych).

3. Gdy już mamy zdefiniowane i zainicjowane wszystkie struktury odpowiadające za menu powinniśmy wywołać procedurę Intuition.SetMenuStrip(). Należy zapamiętać, iż przed zamknięciem okna należy odłączyć menu od okna poprzez użycie procedury Intuition.ClearMenuStrip(). Przy zmianie menu należy także odłączać stare menu od okna i przyłączać nowe.

INICJOWANIE STRUKTUR MENU

Na samym początku zajmiemy się zainicjowaniem struktury Menu:

STRUCT Menu

00 APTR NextMenu ; wskaźnik do następnej struktury Menu odnoszącej się do kolejnej listy menu. Ostatnia struktura Menu w liście tych pozycji głównych musi mieć tę zmienną ustawioną na zero.

04 WORD LeftEdge ; pozioma pozycja lewej krawędzi obszaru traktowanego jako menu.

06 WORD TopEdge ; pionowa pozycja górnej krawędzi obszaru traktowanego jako menu. Aktualnie to pole jest nieużywane gdyż wszystkie menu są ustawiane na samej górze ekranu. Aby zachować kompatybilność z przyszłymi wersjami systemu operacyjnego powinniśmy ustawiać tę wartość na zero.

08 WORD Width ; szerokość obszaru traktowanego jako menu.

0A WORD Height ; wysokość obszaru traktowanego jako menu. Aktualnie to pole jest nieużywane gdyż wszystkie menu posiadają taką samą szerokość i jest ona równa szerokości paska na górze ekranu. Aby zachować kompatybilność z przyszłymi wersjami systemu operacyjnego powinniśmy ustawiać tę wartość na zero.

0C WORD Flags ; znaczniki określające zachowanie się menu. Aktualnie dostępne są tylko dwa różne znaczniki.

0E APTR MenuName ; wskaźnik do zakończonego zerem tekstu określającego nazwę menu. Tekst pojawi się w określonym obszarze na górze ekranu.

12 APTR FirstItem ; wskaźnik do struktury MenuItem dla pierwszej pozycji w liście menu dla tego menu.

16 WORD JazzX ; używane wyłącznie przez bibliotekę Intuition. Należy ustawić na zero.

18 WORD JazzY ; używane wyłącznie przez bibliotekę Intuition. Należy ustawić na zero.

1A WORD BeatX ; używane wyłącznie przez bibliotekę Intuition. Należy ustawić na zero.

1C WORD BeatY ; używane wyłącznie przez bibliotekę Intuition. Należy ustawić na zero.

Znaczniki jakie mogą znajdować się w polu Flags w strukturze Menu:

MENUENABLED - jeżeli ten znacznik jest ustawiony to menu jest dostępne dla użytkownika i może on dokonywać wyboru pozycji z tego menu. Jeżeli ten znacznik nie zostanie ustawiony to menu jest niedostępne dla użytkownika i nie może on dokonać wyboru żadnej pozycji z tego menu. Jeżeli chcemy aby menu było dostępne w momencie jego uruchomienia to możemy ustawić ten znacznik. Menu można w późniejszym czasie uczynić dostępnym lub nie poprzez wywoływanie procedur

Intuition.OnMenu() - udostępnienie menu dla użytkownika oraz

Intuition.OffMenu() - zabronienie dostępu do menu przez użytkownika.

MIDDRAWN - ten znacznik jest ustawiany przez bibliote-

kę Intuition w momencie gdy menu jest wyświetlane dla użytkownika.

A teraz mały przykład zdefiniowania prostej struktury Menu:

```
MyMenu dc.l 0   ; nasze Menu jest pierwszym a zara-
                  zem ostatnim w liście (nie ma żadnych
                  innych menu).
dc.w 0          ; lewa krawędź menu
dc.w 0          ; górna krawędź menu (ignorowana
                  przez intuition - ustawiamy na 0).
dc.w 60         ; szerokość obszaru aktywnego jako
                  menu (w pixlach).
dc.w 0          ; wysokość obszaru menu (ignorowana
                  przez intuition - ustawiamy na 0).
dc.w MENUENABLED
                ; znaczniki - to menu będzie aktywne
dc.l MM_Text    ; wskaźnik do tekstu określającego
                  nazwę menu
dc.l MM_MyItem1 ; wskaźnik do pierwszej struktury
                  MenuItem określającej pierwszą
                  pozycję menu w naszym menu.
dc.w 0,0,0,0    ; wartości JazzX, JazzY, BeatX, BeatY
                  - ustawiamy na 0
MM_Text dc.b 'System',0
```

INICJOWANIE STRUKTUR MENUITEM

Struktura MenuItem określająca daną pozycję menu wygląda następująco:

STRUCT MenuItem

00 APTR NextItem ; wskaźnik do następnej struktury MenuItem określającej kolejną pozycję menu w liście menu. Ostatnia pozycja powinna mieć to pole ustawione na 0.

04 WORD LeftEdge ; pozioma pozycja obszaru, w którym znajduje się dana pozycja menu względem jego lewej krawędzi pobieranej ze struktury Menu dla tej listy menu.

06 WORD TopEdge ; pionowa pozycja obszaru dla danej pozycji względem najwyższej pozycji dozwolonej przez intuition. Najwyższa pozycja jest kalkulowana przez bibliotekę intuition automatycznie na podstawie wysokości użytej czcionki, ilości pozycji ponad nią i tym podobnych. Powinniśmy ustawić na zero aby dana pozycja menu znajdowała się na najwyższej dozwolonej dla niej pozycji.

08 WORD Width ; szerokość obszru zawierającego daną pozycję menu.

0A WORD Height ; wysokość obszaru zawierającego daną pozycję menu.

0C WORD Flags ; znaczniki jakie możemy użyć oraz znaczniki ustawiane przez bibliotekę intuition.

0E LONG MutualExclude ; to pole określa jakie pozycje menu (tylko pozycje określające parametry) mają zostać wyłączone gdy ta pozycja zostanie włączona. Zerowy bit odnosi się do pierwszej pozycji w aktualnej liście, pierwszy do drugiej i tak dalej. Dla przykładu wartość $0000000B (binarnie %00000000000000000000000000001011) wskaże bibliotece intuition, iż ma próbować wyłączyć pierwszą, drugą i czwartą pozycję jeżeli będą to pozycje określające parametry.

12 APTR ItemFill ; wskaźnik do struktury Image (jeżeli znacznik ITEMTEXT jest nie ustawiony) użytej do zastąpienia tekstu elementem graficznym lub do struktury ItemText (jeżeli znacznik ITEMTEXT jest ustawiony) użytej do wprowadzenia tekstu do tej pozycji.

16 APTR SelectFill ; wskaźnik do struktury Image (jeżeli znacznik ITEMTEXT jest nie ustawiony) użytej do zastąpienia tekstu elementem graficznym lub do struktury ItemText (jeżeli znacznik ITEMTEXT jest ustawiony) użytej do wprowadzenia tekstu do tej pozycji, w momencie gdy dana pozycja jest uaktywniana (to pole jest aktywne tylko w przypadku gdy ustawimy znacznik HIGHIMAGE w polu Flags).

1A BYTE Command ; jeżeli użytkownik ma mieć możliwość osiągnięcia menu z klawiatury powinniśmy wskazać w tym polu bibliotece intuition jakiego klawisza ma użyć do uaktywnienia danej pozycji. To pole powinno zawierać kod ASCII pożądanego znaku (należy pamiętać aby włączyć znacznik COMMSEQ w polu Flags).

1B BYTE KludgeFill00

1C APTR SubItem ; wskaźnik do sub-listy menu, która składa się z połączonych ze sobą w listę menu struktur MenuItem. Jeżeli aktualnie wybierana pozycja jest już subpozycją to pole to jest ignorowane przez bibliotekę intuition.

20 WORD NextSelect ; to pole jest ustawiane przez intuition i zawiera specjalną wartość jeżeli więcej pozycji zostało wybranych za jednym razem z tej samej listy menu. W przeciwnym razie zawiera wartość odpowiadającą MENUNULL. Więcej o tym - w dalszej części artykułu. Ustawiamy to pole na MENUNULL podczas inicjalizacji.

Dostępne znaczniki dla struktury MenuItem:

CHECKIT - Ustawiamy ten znacznik jeżeli chcemy aby dana pozycja była pozycją określającą parametry w przeciwnym wypadku będzie to pozycja określająca akcję. Jeżeli ten znacznik jest ustawiony to przed tekstem określającym tę pozycję intuition rezerwuje miejsce na znaczek weryfikacji, który pojawia się gdy jest aktywna i wyłącza się gdy ta pozycja jest nie aktywna. Należy się upewnić czy zarezerwowaliśmy wystarczającą ilość miejsca na znaczek weryfikacji.

CHECKED - Jeżeli dana pozycja jest pozycją określającą parametry (ustawiony znacznik CHECKIT) to znacznik ten jest automatycznie ustawiany gdy użytkownik wybrał ją i uaktywnił a kasowany gdy użytkownik zdeaktywował tę pozycję lub została zdeaktywowana na skutek zadziałania wzajemnego wyłączania poprzez inną pozycję. Możemy ustawić ten znacznik zanim dostarczymy bibliotece intuition gotowe listy do podłączenia jeżeli chcemy aby dana pozycja określająca atrybuty była uaktywniona na starcie.

HIGHITEM - Jeżeli dana pozycja jest aktualnie pod pozycją kursora to ten znacznik jest ustawiany. Aby określić sposób zachowania się danej pozycji w momencie gdy użytkownik najeżdża na nią kursorem powinniśmy ustawić jeden z następujących znaczników:

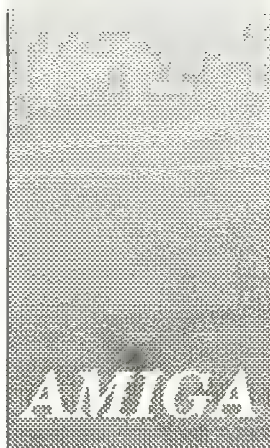
HIGHCOMP - Dokonuje dopełnienia kolorów w aktualnie wybieranej pozycji.

HIGHBOX - Rysuje ramkę wokół obszaru określającego daną pozycję menu.

HIGHIMAGE - Wyświetla grafikę zawartą w strukturze Image lub drukuje tekst pobierany z pola SelectFill w przypadku gdy użytkownik wybiera tę pozycję.

HIGHNONE - Po najechaniu przez użytkownika kursorem na tę pozycję, nie wystąpią żadne zauważalne efekty (jedynie nasz program zostanie o tym poinformowany).

ITEMENABLED - Jeżeli ten znaczni jest ustawiony to dana pozycja jest dostępna dla użytkownika i może dokonać jej wyboru. Natomiast w przypadku nieustawienia tego



znacznika pozycja jest niedostępna dla użytkownika. Jeżeli chcemy daną pozycję udostępnić to możemy użyć procedury Intuition.OnMenu(), w przeciwnym przypadku procedury Intuition.OffMenu().

ITEMTEXT - Jeżeli zawartość pola menu ma być wyświetlana jako tekst to powinniśmy użyć tego znacznika. Pola ItemFill i SelectFill będą traktowane jako pola zawierające wskaźniki do struktur IntuiText. W przypadku nieustawienia tego znacznika pola ItemFill i SelectFill są traktowane jako wskaźniki do struktur Image zawierających graficzne przedstawienie danej pozycji.

COMMSEQ - Jeżeli pozycja ma być dostępna także z klawiatury to powinniśmy ustawić ten znacznik a do pola Command wpisać kod znaku jaki odpowiada za wywołanie tej pozycji.

ISDRAWN - Znacznik ten jest ustawiany przez intuition jeżeli aktualnie wyświetlana jest sub-lista dla danej pozycji. W przeciwnym wypadku znacznik ten jest zerowany.

A teraz przykłąd zdefiniowania struktury MenuItem:

```
MM_MyItem1 dc.l MM_MyItem2
                 ; Wskaźnik do kolejnej struktury
                   MenuItem (kolejna pozycja w liście
                   Menu).
dc.w 0           ; lewa krawędź pozycji względem menu
dc.w 0           ; górna krawędź pozycji (0 określa
                   najwyższą pozycję dostępną dla tej
                   pozycji menu).
dc.w 150         ; szerokość obszaru aktywnego jako
                   pozycja
dc.w 10          ; wysokość obszaru aktywnego
                   jako pozycja
dc w HIGHCOMP+ITEMENABLED+ITEMTEXT+COMMSEQ
                 ; znaczniki określające iż pozycja ma
                   być dostępna, podczas najechania
                   kursorem ma się dokonywać
                   dopełnienie kolorów, tekst ma być
                   użyty do określenia wyglądu pozycji
                   oraz ma być dostępna z klawiatury
dc.l 0           ; nie dokonywane jest wzajemne
                   wyłączanie.
dc.l MI1_IntuiText ; wskaźnik do struktury IntuiText dla
                   tekstu do określenia wyglądu pozycji
dc.l 0           ; SelectFill - ustawiamy na zero gdyż
                   zmiana wyglądu pozycji po najechaniu
                   na nią kursorem odbywa się na
                   zasadzie dopełnienia kolorów
dc.b 'L',0       ; klawisz odpowiedzialny za wywołanie
                   danej pozycji.
dc.l 0           ; ta pozycja nie ma żadnej sub-listy
                   dołączonej
dc.w MENUNULL    ; ustawiamy na wartość MENUNUL -
                   później zmieniane przez bibliotekę
                   intuition.
```

JAK POŁĄCZYĆ I ROZŁĄCZYĆ MENU Z OKNEM

Kiedy już mamy zdefiniowane i zainicjowane odpowiednie struktury Menu i odpowiadające inm struktury MenuItem i wszystko jest połączone razem jedyne co trzeba zrobić to wywołać procedurę INtuition.SetMenuStrip() i menu zostanie podłączone do naszego okna. Należy pamiętać, iż najpierw należy otworzyć okno a dopiero potem podłączyć menu. Aby dokonać zmian w menu aktualnie dołączonym do okna należy to menu najpierw odłączyć od tego okna poprzez wywołanie procedury Intuition.ClearMenuStrip a następnie dokonać zmian i znowu przyłączyć do okna. Jeżeli chcemy zamknąć okno z aktualnie przyłączonym menu, to najpierw musimy usunąć to menu a dopiero potem zamknąć okno. Nieodłączenie menu może spowodować zawieszenie systemu, dlatego zawsze przed zamknięciem okna należy wywołać procedurę Intuition.ClearMenuStrip().

ODBIERANIE KOMUNIKATÓW O OKNACH

W bibliotece Intuition istnieją dwa specjalne znaczniki IDCMP odpowiadające za menu. Są to: MENUPICK i MENUVERIFY. Znacznik MENUPICK oznacza, iż użytkownik wchodził do menu. Mógł on wybrać jedną lub kilka pozycji lub też nie wybrał żadnej i opuścił menu. MENUVERIFY pozwala naszemu programowi na zakończenie poszczególnych czynności przed wyświetleniem menu, lub też można zaniechać wyświetlania menu.

Jeżeli chcemy aby program otrzymywał komunikat o tym iż użytkownik wchodził do menu to musimy w polu IDCMPFlags w strukturze NewWindow ustawić znacznik MENUPICK a następnie możemy obsługiwać komunikaty otrzymywane z portu jako normalne komunikaty IDCMP (na temat IDCMP pisaliśmy w czerwcowym numerze magazyu '64 PLUS 4' w Kąciku Początkującego Kodera).

Kiedy otrzymamy komunikat MENUPICK musimy sprawdzić czy jakakolwiek pozycja została wybrana. Pole Code w komunikacie IntuiMessage będzie zawierało specjalną wartość (Menu Number) określającą numer wybranego menu, na podstawie której określamy numer wybranej pozycji. Jeżeli wartość ta jest równa wartości MENUNULL to nie została wybrana żadna pozycja. Możemy użyć procedury Intuition.ItemAddress() aby określić adres pozycji wybranej przez użytkownika. Do procedury Intuition.ItemAddress() podajemy adres naszego menu oraz numer pozycji otrzymany z komunikatu.

Problem powstaje wtedy gdy użytkownik wybiera kilka pozycji podczas jednego wejścia do menu a nasz program ma dokonać analizy wszystkich. Jednak istnieje bardzo proste rozwiązanie tego problemu: jeżeli zostało wybrana jakaś pozycja to sprawdzamy w niej pole NextSelect i jeżeli wartość jest równa wartości MENUNULL to użytkownik wybrał tylko tę jedną pozycję. W momencie gdy odczytana wartość jest różna od MENUNULL to wartość ta jest numerem kolejnej pozycji i znowu możemy użyć procedury Intuition.ItemAddress() otrzymamy adres następnej pozycji i znowu zprawdzamy pole NextSelect i tak dalej aż do czasu gdy otrzymamy w nim wartość MENUNULL.

Wartość MenuNumber (16 bitów) określająca numer menu składa się z następujących elementów:

c c c c c b b b b b b a a a a a

Bity a (od bitu 0 do bitu 4) określające wartości od 0 do 31 oznaczają, iż zostało wybrane Menu o numerze od 1 do 31 (wartość 0 odpowiada menu numer 1, wartość 1 menu numer 2 i tak dalej...). Wartość 31 jest równa wartości NOMENU - nie wybrano żadnego Menu.

Bity b (od bitu 5 do bitu 10) określające wartości od 0 do 63 oznaczają, iż została wybrana pozycja o numerze od 1 do 63 (wartość 0 odpowiada pozycji numer 1, wartość 1

pozycji numer 2 i tak dalej...) Wartość 63 jest równa wartości NOITEM - nie wybrano żadnej pozycji.

Bity c (od bitu 11 do bitu 15) określające wartości od 0 do 31 oznaczają, iż została wybrana sub-pozycja o numerze od 1 do 31 (wartość 0 odpowiada sub-pozycji numer 1, wartość 1 sub-pozycji numer 2 i tak dalej...) Wartość 31 jest równa wartości NOSUB - nie wybrano żadnej sub-pozycji.

Znacznik IDCMP: MENUVERIFY określa komunikat mówiący, iż użytkownik ma zamiar otworzyć menu (tj. wcisnął prawy przycisk myszki) Dzięki temu menu będzie uaktywnione dopiero w momencie gdy odpowiemy na ten komunikat (Exec.ReplyMsg()). Możemy w ten sposób zaniechć otwierania okna jeżeli kursor myszki znajduje się w innej pozycji niż przez nas pożądana (na przykład otwarcie menu może następować tylko wtedy gdy kursor jest na samej górze ekranu). Gdy otrzymamy komunikat za każdym razem gdy menu ma być otwarte to w polu Code otrzymujemy wartość określającą menu. Jeżeli jest to menu naszego programu (z aktywnego okna) to wartość ta jest równa warości MENUHOT. W przypadku gdy menu należy do innego okna otrzymamy wartość równą wartości MENUWAITING. Gdy chemy zaniechać wyświetlenia menu to przed odpowiedzią na komunikat do pola Code wpisujemy wartość odpowiadającą MENUCANCEL.

PROCEDURY POWIĄZANE Z MENU

Intuition.SetMenuStrip()
SetMenuStrip (Window, Menu)
A0 A1
Procedura dokonuje przyłączenia menu do okna. Należy pamiętać o wcześniejszym otworzeniu okna jeszcze przed przyłączeniem menu.
Wejście:
Window - wskaźnik do struktury Window otrzymywanej z procedury Intuition.OpenWindow().
Menu - wskaźnik do listy menu wskazujący na pierwszą strukturę Menu.

Intuition.ClearMenuStrip()
ClearMenuStrip (Window)
A0
Procedura dokonuje odłączenia menu od okna. Należy pamiętać aby zawsze odłączyć menu od okna przed zamknięciem tego okna lub zmianami w liście menu.
Wejście:
Window - wskaźnik do struktury Window otrzymywanej z procedury Intuition.OpenWindow().

Intuition.ItemAddress()
ItemAddress (Menu, MenuNumber)
A0 D0
Procedura oblicza adres pod jakim znajduje się struktura MenuItem (lub struktura Menu) dla podanej wartości (wartość pobierana z komunikatu IntuiMessage lub ze struktury MenuItem-NextSelect).
Wejście:
Menu - wskaźnik do listy menu wskazujący na pierwszą strukturę Menu.
MenuNumber - wartość określająca numer menu, pozycji i sub-pozycji pobierana z komunikatu IntuiMessage lub z pola NextSelect ze struktury MenuItem.

Intuition.OffMenu()
OffMenu (Window, MenuNumber)
A0 D0
Procedura blokuje dostęp do sub-pozycji, pozycji lub całego menu. Użytkownik nie będzie mógł dokonać wyboru tego menu
Wejście:
Window - wskaźnik do struktury Window otrzymywanej z procedury Intuition.OpenWindow().
MenuNumber - wartość określająca numer menu, pozycji i sub-pozycji pobierana z komunikatu IntuiMessage lub z pola NextSelect ze struktury MenuItem, lub obliczana przez nas na podstawie podanego wcześniej ustawienia bitów. Jeżeli określimy całe Menu użytkownik nie będzie miał dostępu do żadnej pozycji ani sub-pozycji w tym menu.

Intuition.OnMenu()
OnMenu (Window, MenuNumber)
A0 D0
Procedura odblokowuje dostęp do sub-pozycji, pozycji lub całego menu. Użytkownik będzie mógł dokonać wyboru tego menu
Wejście:
Window - wskaźnik do struktury Window otrzymywanej z procedury Intuition.OpenWindow().
MenuNumber - wartość określająca numer menu, pozycji i sub-pozycji pobierana z komunikatu IntuiMessage lub z pola NextSelect ze struktury MenuItem, lub obliczana przez nas na podstawie podanego wcześniej ustawienia bitów. Jeżeli określimy całe Menu użytkownik będzie miał dostęp do wszystkich pozycji i sub-pozycji w tym menu.

Marcin "Duddie" Dudar



AMIGA

GRACZ DOSKONAŁY

❑ Another World
Kilka kodów do kolejnych poziomów (aby je wpisać należy nacisnąć najpierw 'C'):
1 EDJI 2 HICI 3 FLLD 4 LABC 5 CCAL 6 EDIL 7 KCIJ 8 FADK 9 LPCI
10 LAEA 11 LFEK

❑ Atomic Robo Kid
Podczas wyświetlania strony tytułowej wpisz TUESDAY 14TH po naciśnięciu FIRE ukaże się na ekranie menu umożliwiające wybranie różnych ułatwień (na przykład nieskończone życia itp).

❑ Battle Island
Kody do kolejnych poziomów:

00 FIRST	01 GHOST	02 GAMMA	03 MARSS	04 EAGLE	05 METAN	06 FOTON
07 POLAR	08 TIGER	09 SNAKE	10 ZENIT	11 DONNN	12 VESTA	13 OXXID
14 DEMON	15 GIANT	16 CONRA	17 PHASE	18 EXOTY	19 MOUNT	20 FIGHT
21 RUSTY	22 FIFTH	23 VESUV	24 MAGIC	25 SPACE	26 VALEY	27 TESTY
28 TERRA	29 SLAVE	30 NEVER	31 RIVER	32 EUROP	33 STORM	

❑ Beverly Hills Cop
Aby uzyskać pewne ułatwienie wpisz MELLIE podczas wyświetlania planszy z wyborem stopnia trudności gry.

❑ Beyond the Gates
Wpisanie THE END OF TIME DRAWS NEAR wywoła edytor, dzięki któremu można dowolnie modyfikować swój aktualny status.

❑ Bionic Commando
Wyjdź z pierwszego poziomu dokładnie wtedy, gdy licznik czasu będzie wskazywał 1. Otrzymasz wtedy 9 żyć oraz bardzo dużo dodatkowych punktów.

❑ Brat
Oto kilka kodów do gry:
2 MIHEMOTO 3 SASUTOZO 4 SUMATZEE 5 NOKITAGO 6 ITSANONO 7 MOZIMATO
8 HOZITOMO 9 MOKITEMO 10 ZUMOHATO 11 CHANASTU 12 NAGAITSU

❑ Capone
Aby uzyskać nieśmiertelność strzel dwukrotnie w maszt umieszczony przed budynkiem poczty.

❑ Chariots of Wrath
Gdy pokazuje się napis, aby nacisnąć FIRE w celu rozpoczęcia gry, zamiast niego skieruj joystick do góry.

❑ Chase H.Q.
Przytrzymaj lewy przycisk myszy oraz przycisk fire i wpisz GROWLER. Nacisnij teraz na dłuższą chwile klawisz T. Teraz pod koniec każdego poziomu naciśnij klawisz space, oprócz tego dzięki klawiszom X oraz Z możesz łatwiej sterować. Spróbuj także po zniknięciu obrazka tytułowego dość szybko uderzać w klawisz space (podczas ładowania). Dzięki temu twój samochód bedzie mógł osiągnąć prędkość do 1000km/h.

❑ Chuck Rock
Naciśnij (na stronie tytułowej) klawisz ESC i wpisz FAST AINT THE WORD lub UNCLE SAMS lub ITS FAIRY BOWBELZ lub SHE LOVES CLEANING WINDOWS lub LIFE IS MY DREAM, aby uzykać nieskończoną energię.

AMIGA

❑ Crackdown

Podczas gry wciśnij klawisz pauzy i wpisz SMURF. Od tej chwili naciśnięcie F1 i F2 znacznie poprawi twoją sytuację.

❑ Crystal Hammer

Gdy zaczynasz grę przytrzymaj jeden przycisk myszy i kliknij na drugi zanim pokaże się piłka. Wystartujesz od poziomu 26.

❑ Defender II

Kilka ciekawych haseł:
GOATY - niewidzialność
RAVEN - od tej chwili jest kilka dodatkowych klawiszy:
I - włączenie/wyłączenie niewidzialności
N - następny poziom
D - 23 poziomy do przodu
INCAS - daje nieco lepszą broń
ANDES - energia wzrasta przy każdym strzale
Na koniec jeszcze kilka kodów do gry:

1) START	5) FLOYD	9) FURRY	13) BEAST	17) LEMAC
21) ZIPPO	25) LASER	29) DEFAD	33) MAGOG	37) FUNKY
41) DONKY	49) KANJI	53) IRATA	57) NEURO	61) STOAT

❑ Dr Plummet

F1 do następnego poziomu, F2 niewidzialność.

❑ Drivin Force

Kliknij myszą na obie literki 'i' w napisie DRIVIN na głównej planszy, a podczas gry "wylecenie" z toru stanie się niemożliwe.

❑ Dyter-07

Podczas wczytywania z dysku wpisz GIBB. Spowoduje to włączenie specjalnego trybu gry, w którym możesz używać dwóch dodatkowych klawiszy 'W' aby uzyskać dodatkową broń, a 'S' aby otrzymać ponownie pełny zapas energii.

❑ Elephant Antics

Wpisz ITCHY ARSEHOLES na stronie tytułowej, aby uzyskać nieśmiertelność.

❑ Elf

Wpisanie CHOROPOO powinno znacznie ułatwić grę.

❑ Empire Strikes Back

Przytrzymaj klawisz HELP i wpisz XIFARG ROTKEV. Warto teraz wypróbować klawisze cyfr oraz L, C i D.

❑ Enchanted Lands

Wpisz TCB RULES FOREVER podczas trwania czołówki. Ekran powinien zamigać. Od tej chwili F3 spowoduje wejście do trybu edycji, a F2 poprzedzone SPACE przeniesie na koniec danego poziomu.

❑ Eye of Horus

Gdy ukaże się napis PRESS FIRE TO START wpisz SPAM aby uzyskać niewidzialność, poza tym nie trzeba więcej szukać kolorowych kluczy do wind.

❑ F-29 Retaliator

Wpisz się jako DIDYMEN, kliknij na ikonę COLONEL i wciśnij RETURN.

❑ Flying Shark

Podczas wyświetlania ekranu z punktami wpisz HSC lub KDJ co spowoduje znacznie ułatwienia w grze.

❑ Forgotten Worlds

Wpisz ARC w czołówce i naciśnij HELP. Od tego momentu podczas gry można używać S, aby wejść do sklepu lub N aby przejść do następnego poziomu.

❑ Ghostbusters II

Przytrzymaj ALT CTRL 5 U gdy pokazuje się napis Activision i naciśnij FIRE. Oprócz zwykłej nieśmiertelności możemy być także pewni, iż nie zerwie się lina na pierwszym poziomie.

AMIGA

❑ Hacker II - The Doomsday Papers

Kilka ciekawych haseł:
COVER H2SC - zakończenie gry
TITLE H2SC - strona tytułowa
DEMO H2SC - przyspiesza grę, ale także ułatwia.

❑ Hard Drivin

Rozwiń największą możliwoą prędkość i naciśnij N. Od tej chwili możesz kontynuować jazdę z tą predkościa ze "wspomaganiem kierownicy", a także będąc niewidzialnym oraz mając nieskończony zapas czasu.

❑ Hillsfar

Gdy próbujesz otwierać zamki możesz sobie to nieco ułatwić naciskając klawisz F.

❑ Insanity Flight

Aby móc przejść do dowolnego poziomu trzeba nacisnąć równocześnie oba klawisze myszy oraz przycisk FIRE joysticka. Wystarczy nacisnąć jeszcze klawisz L.

❑ International Karate

Kilka przydatnych podczas gry haseł:
FREZ - zatrzymanie gry,
PAC - przez ekran przebiegnie PAC MAN,
FISH - ryba wyskoczy z wody,
BIRD - przez ekran przeleci ptak,
PERI - z wody wynurzy sie peryskop.

❑ James Pond

Wpisz JUNKYARD i naciśnij RETURN. Od tej chwili D otwiera wszystkie zamki, kolejne klawisze od Z do M powodują przeniesienie do odpowiednich poziomów.

❑ Judge Dredd

Wpisz się do komputera jako DREDD i wpisz BRUCKEN PLAYING HEROQUEST i wyjdź z systemu. Od tego momentu klawiszem HELP możesz przechodzić do następnego poziomu.

❑ Kick Off

Aby obronić każdy strzał na bramkę wystarczy przytrzymać przycisk FIRE. Strzał będzie zablokowany.

❑ Kick Off 2

Naciskaj R gry będziesz musiał obronić karnego. W ten sposób zobaczysz, gdzie będzie oddany strzał.

❑ Killing Cloud

Kilka haseł do kolejnych poziomów:
2 - A66TG7EZ   3 - 2WWTQ7E3   4 - QXX6G6EB   5 - 3336RWE3

❑ LED Storm

Wpisz DAVID BROADHURST WANT TO CHEAT. To wystarczy, aby gra stała się bardzo prosta.

❑ Legend of the Lost

Hasła do kilku poziomów: 2) STONES 3) LADDER 4) ESCAPE 5) LAVA 6) FINALE

❑ Mickey Mouse

Wpisz 61315688 aby uruchomić specjalny tryb gry, po którego uruchomieniu działają dodatkowo następujące klawisze:
F2 - otwiera drzwi do kolejnych gier,
F3 - rozpoczyna walkę z wiedźmą,
F4 - ponownie wypełnia twój pistolet na wodę.

❑ Narco Police

Bardzo przydatne podczas gry hasła:
MUNICION - zwiększenie ilości amunicji o 200,
NOAMETZ# - wyłącza karabiny maszynowe w sektorze o podanym numerze (#),
NOCAMZ# - włącza karabiny maszynowe w sektorze o podanym numerze (#),
NOENEMIG - zabija przeciwników,
COMENZAR - powoduje powstanie zabitych przeciwników,
ABRIR - otwiera dowolne drzwi.

AMIGA

❑ Pegasus

Kody do gry:
SCREECH - 10-19 DRAGONFLY - 20-29 BEEBOP - 30-39 CELE:

❑ Platoon

Wpisz HAMBURGER na stronie tytułowej. Od tej chwili działają następujące klawisze (podczas gry):
F1 - zaczyna od momentu w którym jesteś,
F2 - przenosi do miejsca z materiałami wybuchowymi,
F3 - przenosi na most,
F4 - przenosi do wioski,
F5 - nieśmiertelność.

❑ Prince of Persia

Przy pomocy jednej z kombinacji CAPS J lub CAPS L możesz przejść do następnego poziomu.

❑ Railroad Tycoon

Dodatkowe pieniądze można otrzymać naciskając klawisze SHIFT oraz 4.

❑ Revelation

Kody do kolejnych poziomów:
10 'Sirens' 20 'Loader' 30 'Player' 40 'Result' 50 'Dollar'
60 'Change' 70 'Finger'

❑ Sentinel

Kilka kodów:

3) 70511958	8) 16257084	10) 43428170	11) 40556356	18) 68627185
19) 55742564	36) 97578966	50) 61185004	61) 26060764	66) 68657887
80) 18452261	103) 36873128	127) 06567770	150) 92856449	169) 16788555
196) 72957558	225) 38546570			

❑ Toki

Po wpisaniu KILLER podczas gry ekran zacznie migać i włączy się nieco ułatwiony tryb gry. Spróbuj też wpisać KILLER po raz drugi, co spowoduje pokazanie dość śmiesznego efektu. Po pierwszym wpisaniu można też korzystać z klawiszy funkcyjnych do zmiany poziomu.

❑ Trailblazer

Na stronie tytułowej przytrzymaj HELP, I, 1 oraz 2 i wpisz CHEAT. Od tego momentu podczas gry działają dodatkowe klawisze:
F3 - usuwa grawitację,
F5 - zmienia kształt piłki,
F6 - zmienia piłki innych graczy,
HEAR - włącza śmieszny efekt.

❑ Venus - The Flytrap

Oto kilka kodów do kolejnych poziomów:
02 - The Frozen Wastes - MANTIDAS
03 - The Dead City - CICADAS
04 - Wood World - PSYLLIDS
05 - The Caverns - PIERIDS
06 - Death Valley - SATYRID
07 - The Creeping Swamp - LYCAENID
08 - Tech World - PYRALID
09 - Translucent Plain - NOCTUID
10 - The Stygian Creek - NO PASSWORD!
Oraz kilka ciekawych haseł:
JUPITER - nieskończone życia,
MARS - dostęp do wszystkich broni,
SATURN - hasło do wypróbowania!
PLUTO - nieskończona broń,
MERCURY - ciągła umiejętność latania.

❑ Wizball

Zatrzymaj grę i wpisz RAINBOWT. Spowoduje to wprowadzenie kilku ułatwień do gry.

AMIGA

AMOS THE CREATOR

- czyli piszemy własną grę! Cz. 2

Tak jak obiecałem w poprzednim odcinku kursu AMOS'a, w tym miesiącu zaznajomimy się z czymś więcej niż tylko najprostszymi komendami w rodzaju Draw i Plot. Dzisiaj poznamy komendy bardziej zaawansowane, pozwalające na większą swobodę i elastyczność w operowaniu grafiką. Dowiemy się także w jaki spsób manipulować ekranami graficznymi AMOS'a. A więc zaczynamy!

Po poznaniu komend Draw i Plot, oraz po przeanalizowaniu poprzedniego programu, którego nota bene możecie także używać jako pomoc w rozwiązywaniu trudnych zadań matematycznych, możemy przejść do rzeczy trudniejszych. Kolejną instrukcją graficzną jest

Box Xp,Yp To Xk,Yk.

Instrukcja ta służy do rysowania prostokąta. Współrzędne Xp oraz Yp oznaczają lewy górny róg figury, natomiast Xk,Yk to współrzędne prawego dolnego rogu. Oznacza to, iż długość poziomego boku wynosi Xk-Xp, natomiast długość boku pionowego jest równa Yk-Yp, co jest oczywiste, ale być może mniej oczywisty jest wynikający z tego fakt, że muszą zostać zachowane dwie nierówności Xk = Xp, oraz Yk = Yp . Gdy te dwie nierówności będą spełnione, wszystko będzie w porządku. Jeżeli Xp będzie równe Xk, a Yp będzie takie samo jak Yk to na ekranie ujrzymy punkt o współrzędnych Xp,YP (albo oczywiście Xk,Yk co na jedno wychodzi).

Kolejna instrukcja jest już bardziej skomplikowana. Tą instrukcją jest:

Polyline X1,Y1 To X2,Y2 To ... To Xn,Yn.

Jedynym ograniczeniem długości tej instrukcji jest długość linii w AMOS'ie równa maksymalnie 255 znakom. Instrukcja Polyline jest to zwielokrotnienie instrukcji Draw. Na przykład jedna instrukcja

Polyline 10,10 To 20,20 To 30,30

odpowiada dwóm instrukcjom:

Draw 10,10 To 20,20 : Draw 20,20 To 30,30

jak sądzę wszyscy wiemy co jest krótsze i bardziej elegenckie (a co najważniejsze - szybsze). W miarę możliwości należy korzystać z tej komendy w zastępstwie całej masy komend Draw.

Do rysowania okręgów służy instrukcja

Circle Xs,Ys,R.

Jej parametry to Xs,Ys - czyli środek naszego okręgu, oraz R - czyli promień. Należy pamiętać, że jest to jedna z wolniejszych instrukcji i nie odnosi się do niej zmiana rodzaju linii. Jak zwykle Xs oraz Ys można pominąć, parametry te zostaną zastąpione współrzędnymi położenia kursora graficznego. Należy pamiętać, że w wyniku wykonania poprzedniej instrukcji współrzędne kursora graficznego przyjmują wartość taką jak ostatnia zadana współrzędna. Tak więc wynikiem programu:

Polyline 0,0 To 100,100 To 50,50 : Circle „20

będzie narysowanie dwóch linii 0,0 do 100,100 oraz 100,100 do 50,50 i okręgu o środku w punkcie 50,50 i promieniu równym 20.

Aby uzyskać elipsę należy użyć innej instrukcji:

Ellipse X,Y,a,b

- gdzie X,Y - środek symetrii elipsy, natomiast a - połowa osi małej, b - połowa osi wielkiej. Analogicznie jak przy Circle oraz wielu innych instrukcjach graficznych X,Y może zostać pominięte i zastąpione automatycznie przez położenie kursora graficznego.

Bardzo ciekawym usprawnieniem komend Box, Polyline oraz Draw jest instrukcja

Set Line rodzaj

Komenda ta ustala rodzaj linii. Wzór linii definiuje się za pomocą wartości "rodzaj", którą najlepiej jest podać w formie słowa zapisanego w postaci binarnej (słowo ma 16 bitów). Standardowa linia byłaby zdefiniowana w taki sposób:

Set Line %1111111111111111

- należy pamiętać o "procentach" na początku naszej wartości.

Aby uzyskać linię przerywaną co punkt należy użyć komendy w ten sposób:

Set Line %1010101010101010

jak już chyba widać tworzymy po prostu "wzorek" z zer i jedynek gdzie zera to puste miejsca, a jedynki - miejsca w których pojawi się linia. Wzór linii zostanie załadowany do rejestru BitBDat blittera, ale z poziomu AMOS'a nie musimy się o to troszczyć.

Przedstawię teraz kilka instrukcji pozwalających nam na wypełnianie rozmaitych powierzchni w różny sposób. Pierwsza z nich to:

Bar Xp,Yp To Xk,Yk.

Jest to komenda, która działa dokładnie tak jak Box , ale z jednym małym wyjatkiem - narysowany prostokąt jest wypełniony. Tak samo jak w Box , Xp,Yp to współrzędne lewego górnego rogu, natomiast Xk,Yk - dolnego, prawego. I tak samo jak w Box, należy uważać aby lewy dolny nie stał się nagle prawym górnym rogiem. Kolejną analogią jest

Polygon X1,Y1 To X2,Y2 To ... To Xn,Yn.

Jak łatwo się domyślić jest to instrukcja analogiczna do Polyline, z tym że powstały wielokąt jest wypełniony. Zapomniałem dodać, iż zarówno w Polyline jak i Polygon można pominąć pierwszą wartość, jak zwykle używając kursora graficznego. Zapis wygląda wtedy np. tak:

Polygon , To 36,48 To 10,25.

I wreszcie podstawowa, fundamentalna instrukcja służąca do wypełniania określonych obszarów:

Paint X,Y,tryb.

X,Y to dowolnie wybrany punkt w środku wypełnianego obszaru, natomiast "tryb" określa sposób rozpoznawania granicy obszaru wypełnianego. Dla wartośći "tryb" równej zero granicą wypełnianego obszaru będzie kolor obrysu, natomiast dla wartości "tryb" równej jeden - granicą będzie dowolny kolor różny od koloru farby (ustawiany instrukcją Ink).

Nie musimy jednak ciągle wszystkiego wypełniać kolorem tła. Byłoby to poważne ograniczenie naszych możliwości, a takich przecież nie ma w AMOS'ie. Instrukcja pozwalająca rozwiązać ten problem ma postać:

Set Pattern wzór

Ustawienie wartości "wzór" na zero nie powoduje żadnych zmian - zero to po prostu jednolite wypełnienie

kolorem atramentu. Jeżeli wartość "wzór" jest większa od zera to wybierany jest jeden z 34 zaprogramowanych wzorów ze zbioru Mouse.ABK. Numeracja zaczyna się od czwórki, dlatego iż pierwsze trzy są wykorzystywane jako wzór wskaźnika myszy. W przypadku gdy wartość "wzór" jest mniejsza od zera to zaczyna on określać jeden ze sprite'ów. Jeżeli dajmy na to wzór=-5 to określa on piątego sprite'a, czyli numer sprite'a jest równy wzór * (-1). Jeżeli sprite jest dwukolorowy to nie ma kłopotu - wzór sprite'a po prostu nałoży się na kolor tła.

Kłopoty zaczynaja się gdy sprite jest wielokolorowy, wtedy kolory jego rysunku oraz kolory atramentu są poddawane logicznemu AND, natomiast kolor tła - operacji logicznej OR. W tym przypadku aby zachować naturalne kolory sprite'a musimy ustalić kolory tła na jego kolor zerowy, a atramentu na trzydziesty pierwszy.

Aby rysowane przez nas figury posiadały (lub nie) obrys należy użyć instrukcji

Set Paint n

gdzie n jest znacznikiem - może przyjmować wartość zero lub jeden. Dla wartości n równej jeden, obrys zostaje włączony, natomiast dla n równego zero obrys zostaje wyłączony. Instrukcja Set Paint 0 wyłącza obrys, a instrukcja Set Paint 1 włącza go.

AMOS posiada różne tryby rysowania i nakładania grafiki, jego możliwości nie kończą się na zwykłym nakładaniu grafiki na tło. Do zmiany trybu wyświetlania grafiki służy instrucja

Gr Writing n.

Należy uważać przy ustawianiu trybu rysowania grafiki, gdyż w liczbie n ważne są tylko pierwszw trzy bity, maksymalna jej wartość to siedem (włączone są wszystkie tryby kreślenia grafiki.

Pierwszy tryb - JAM1 jest trybem standardowym, normalnym - bit zerowy musi być ustawiony na zero. Grafika jest rysowana normalnie.

Drugim trybem jest JAM2 - włącza się go nadając wartość jeden dla zerowego bitu. Powoduje, iż nowa grafika jest nakładana na poprzednią. Aby dokładnie wiedzieć jak działa ten tryb radzę wykonać kilka eksperymentów.

Kolejny tryb o nazwie XOR (nazwa od operacji logicznej, którą wykonuje), jest ustawiony wtedy gdy bit pierwszy będzie miał wartość jeden. Tak jak wskazuje na to jego nazwa poddaje grafikę nakładaną - operacji logicznej XOR z tłem, wygląda to tak, że każdy punkt obrazu nakładanego będzie się w sposób widoczny odcinał od otoczenia. Trzeba użyć kilka razy tej opcji aby przekonać się jak ona działa. Bardzo ciekawą cechą tego trybu jest fakt, iż po powtórnym nałożeniu grafiki w to samo miejsce (oczywiście TEJ SAMEJ grafiki) uzyskujemy obraz oryginalny taki jak przed obydwiema operacjami. Drugie nałożenie grafiki "neutralizuje" efekty działania operacji.

Ostatni już tryb graficzny możliwy do uzyskania za pomocą instrukcji Set Paint to tryb INVERSEVID - ustawiamy go przez drugi bit - aby tryb był włączony, bit drugi musi mieć wartość jeden. Tryb ten powoduje "odwrócenie" kolorów grafiki - kolor atramentu staje się kolorem tła i odwrotnie - jest to inversja obrazu. Zawsze warto dokładnie poeksperymentować aby wiedzieć jak działa dana opcja i do tego zachęcam w przypadku tego trybu. Opisanie wyników jego działania jest praktycznie niemożliwe i dlatego trzeba samemu sprawdzić co się dzieje po jego użyciu.

Powyżej wspomniałem o jakichś bitach, pierwszym, drugim ,zerowym... - jak ich używać w praktyce? Jest to tylko troszkę skomplikowane. Podstawowa wiadomość dla użytkownika brzmi - tryby można prawie dowolnie mieszać, tzn. tryb JAM2 może być użyty z INVERSEVID, ale co jest chyba oczywiste nie można na raz użyć trybu JAM1 oraz JAM2.

Jak już wspomniałem tryby ustawiamy używając instrukcji

AMIGA

Gr Writing n

"n" ustawiamy następująco: dla normalnego trybu - JAM1, i bez włączonego żadnego innego trybu mamy n=0. Aby włączyć tryb JAM2 dodajemy jeden. Aby do tego dołączyć tryb XOR musimy dodać jeszcze dwa, a jeżeli chcemy na dodatek tryb INVERSEVID musimy dodać cztery. Tak więc instrukcja np. Gr Writing 4 ustala tylko tryb INVERSEVID oraz (automatycznie) tryb JAM1, natomiast Gr Writing 1+2+4 ustala tryby JAM2, XOR oraz INVERSEVID. Tryby te jeśli chodzi o tekst dotyczą tylko i wyłącznie instrukcji Text.

Czasem nie chcemy mieć grafiki na całym ekranie, a dokonanie ograniczenia pola widzenia środkami konwencjonalnymi jest bardzo trudne, wtedy w sukurs przychodzi nam bardzo ciekawa komenda

Clip Xlg,Ylg To Xpd,Ypd.

Xlg oraz Ylg to współrzędne lewego górnego rogu nowego zmniejszonego ekranu (no. niekoniecznie zmniejszonego, gdyż można podać wartości z poza ekranu), natomiast Xpd i Ypd to współrzędne prawego dolnego rogu nowego obszaru. Podanie samego Clip bez współrzędnych spowoduje powrót do starych wartości.

AMOS dopuszcza maksymalnie aż osiem niezależnych od siebie ekranów graficznych o rozdzielczości i liczbie kolorów zdefiniowanej całkowicie przez użytkownika. Po uruchomieniu AMOS'a, ukazuje się nam podstawowy ekran graficzny o rozdzielczości 320 na 256 punktów i szesnastu kolorach. Parametry te można w każdej chwili zmienić. Do definiowania i zastępowania ekranów graficznych nowymi służy instrukcja

Screen Open nr,szer,wys,lkol,tryb.

"nr" to numer otwieranego ekranu, może przyjmować wartości od zera do siedmiu i jest ustawiany dowolnie. Ekran standardowo ukazujący się zaraz po włączeniu AMOS'a ma numer zero i dlatego podając ten numer możemy go usunąć. Następny parametr komendy - "szer" to szerokość ekranu w punktach. Przy podawaniu szerokości ekranu nie należy się sugerować rozdzielczością - część ekranu może wcale nie być widoczna, i tak będą w niej działać wszelkie operacje graficzne. "wys" - jest to oczywiście wysokość ekranu, przy jej podawaniu nie należy brać pod uwagę rozdzielczości tak jak w przypadku "szer". Parametr "lkol" to liczba dostępnych kolorów na otwieranym przez nas ekranie graficznym. Dopuszczalne liczby kolorów to 2, 4, 8, 16, 32, 64 (tryb Extra Half Bright), 4096 (tryb Hold And Modify). Ostatni parametr o nazwie "tryb" to po prostu tryb graficzny naszego ekranu. Parametr ten przyjmuje wartość Lowres (dla rozdzielczości poziomej równej 320) i Hires (dla rozdzielczości poziomej równej 640). Tak więc np. instrukcja

Screen Open 0,640,256,8,hires

powoduje zamianę standardowego ekranu AMOS'a na ekran o większej rozdzielczości i mniejszej liczbie kolorów. Standardowemu ekranowi AMOS'a odpowiada komenda

Screen Open 0,320,256,16,Lowres.

Warto pamiętać, że ekrany graficzne zajmują dużo pamięci (także ich część niewidoczna!) i warto znać ilość pamięci, która jest im potrzebna a którą podaję:

AMIGA

Rozdzielczość	Liczba kolorów	Ilość potrzebnej pamięci
320 * 256	dwa	8kb
640 * 256	dwa	16kb
320 * 256	cztery	16kb
640 * 256	cztery	32kb
320 * 256	osiem	24kb
640 * 256	osiem	48kb
320 * 256	szesnaście	32kb
640 * 256	szesnaście	64kb
320 * 256	trzydzieści dwa	40kb
Standardowy ekran AMOS'a		
320 * 256	sześćdziesiąt cztery	48kb
tryb EXtra Half Bright		
320 * 256	4096	48kb
tryb Hold And Modify		

Komendą bardzo przydatną w pewnych wypadkach jest
Screen Close nr.
Komenda ta powoduje zamknięcie i usunięcie ekranu graficznego o numerze "nr". Następna komenda dotyczy uaktualniania ekranu. Jest to instrukcja o nazwie
Auto View on/off.
Po włączeniu AMOS'a komenda ta jest ustawiona na ON. Jeżeli użyjemy komendy w ten sposób:
Auto View off
to wszystko co się dzieje na ekranie przestanie być uaktualniane i np. otwarcie nowego ekranu nie spowoduje żadnego efektu, tak samo narysowanie na ekranie np. linii prostej nie spowoduje żadnej reakcji na nasze polecenie. Lecz brak reakcji jest tylko złudzeniem, komputer wykonuje nasze polecenia w pamięci, nie przedstawiajac wyników na ekranie. Można teraz użyć komendy View aby zawartość ekranu została odświeżona. Na ekranie pojawi się nowo zadeklarowany ekran oraz kreska. Jeżeli napiszemy "Auto View on" to zawartość ekranu będzie aktualizowana po każdym użyciu komendy związanej z grafiką. Komenda Default robi porządek z naszymi wszystkimi ekranami (zamyka wszystkie otwarte ekrany!) i otwiera ekran standardowy.
View - jak już wspomniałem komenda ta służy do odświeżania ekranu wtedy gdy użyto instrukcji Auto View off.
Gdy chcemy wyczyścić ekran musimy użyć komendy
Cls kl,Xlg,Ylg To Xpd,Ypd.
Komenda użyta bez żadnych parametrów czyści cały ekran i wszystkie okna kolorem tła. Kolor tła można zastąpić przez dowolny kolor wybrany przez użytkownika i podany przez parametr "kl". Komendą Cls można także wyczyścić dowolny prostokąt na ekranie nie naruszając reszty obrazu. Do takiego właśnie działania służą parametry podane po "kl". Xlg oraz Ylg to współrzędne lewego górnego rogu czyszczonego obszaru, natomiast Xpd i Ypd to współrzędne prawego dolnego rogu. Myślę, że na dziś wystarczy o ekranach graficznych, w następnym odcinku kursu AMOS'a poznamy jeszcze całe mnóstwo operacji jakie można przeprowadzić na ekranach graficznych, dowiemy się co to takiego banki danych AMOS'a i wiele innych bardzo ciekawych rzeczy. Jak co miesiąc tak i dziś podaję program do analizy. Jest to krótki programik demonstracyjny, program pokazuje obiekt narysowany (i obracany) techniką wektorową (z tym, że jest to prosty obiekt i bardzo prosta wektorówka tzw. "drucianka"). Wersja źródłowa oraz tekstowa tego programu znajduje się na Public Domain Pack'u nr. 20 z sierpnia, więc zamiast mozolnie wklepywać program, wystarczy zaopatrzyć się w PDP-Amiga z Sierpnia.

```
Rem
*******************************************************
Rem ****** Prezentacja prostej grafiki wektorowej *****
Rem ***** dla czytelnikow pisma 64+4/Amiga     *****
Rem ***********        autor - Amber        *************
Rem
*******************************************************
Dim IKSY(7) : Dim IGREKI(7) : Dim ZETY(7) : Dim X(7) :
Dim Y(7)
Screen Open 1,640,256,4,Hires
Double Buffer
Autoback 0
        XO=0 : YO=0 : ZO=-100
        XE=0 : YE=0 : ZE=0
        A#=0 :
For I=0 To 7
        Read XD
        Read YD
        Read ZD
        IKSY(I)=XD
        IGREKI(I)=YD
        ZETY(I)=ZD
Next
For ZE=-100 To 200
S#=Sin(A#)
C#=Cos(A#)
For I=0 To 7
XN=IKSY(I)*C#-ZETY(I)*S#
YN=IGREKI(I)
ZN=IKSY(I)*S#+ZETY(I)*C#
XA=XN
YA=YN*C#-ZN*S#
ZA=YN*S#+ZN*C#
XW=XA*C#-YA*S#
YW=XA*S#+YA*C#
ZW=ZA
 X(I)=(XW*(ZE-ZO))/(ZW-ZO)+320
 Y(I)=(YW*(ZE-ZO))/(ZW-ZO)+128
Next
Screen Swap
Cls 0
Draw X(0),Y(0) To X(1),Y(1)
Draw X(1),Y(1) To X(2),Y(2)
Draw X(2),Y(2) To X(3),Y(3)
Draw X(3),Y(3) To X(0),Y(0)
Draw X(0),Y(0) To X(4),Y(4)
Draw X(1),Y(1) To X(5),Y(5)
Draw X(2),Y(2) To X(6),Y(6)
Draw X(3),Y(3) To X(7),Y(7)
Draw X(4),Y(4) To X(5),Y(5)
Draw X(5),Y(5) To X(6),Y(6)
Draw X(6),Y(6) To X(7),Y(7)
Draw X(7),Y(7) To X(4),Y(4)
A#=A#+0.2
Next
Data -25,-25,25,25,-25,25,25,25,25,-25,25,25
Data -25,-25,-25,25,-25,-25,25,25,-25,-25,25,-25
```

Michał "Amber" Gosztyła

REQTOOLS LIBRARY CZ.3

AMIGA

Dzisiejszy odcinek naszych zmagań z biblioteką ReqTools poświęcamy przede wszystkim funkcji rtFileRequest

rtFileRequestA
Wartość = rtFileRequestA (Req, Nazwa, Tytuł, TagList)
D0 A1 A2 A3 A0

Funkcja rtFileRequestA pobiera katalogi i zbiory z dysku umożliwiając użytkownikowi wybór właściwego zbioru. Funkcja ta może także służyć do pobierania samego katalogu.

'Nazwa' powinna wskazywać na bufor o wielkości przynajmniej 108 znaków. Jeżeli do tego bufora wpiszemy nazwę danego zbioru przed wywołaniem requestera to zostanie ona wyświetlona w polu nazwy zbioru w oknie requestera. Przy powrocie do programu prawdopodobnie zostanie zmieniona.

Poprzez ustawianie odpowiednich znaczników można doprowadzić do zainicjowania struktury Hook, która jest funkcją dostarczoną przez użytkownika i będzie wywoływana przez procedurę rtFileRequestA(). Nasza procedura będzie wywoływana z parametrami przekazywanymi w rejestrach a0/a1/a2. Rejestr a0 będzie zawierał wskaźnik do naszej struktury Hook, w rejestrze a1 będzie umeszczony wskaźnik do struktury Req dla requestera, który wywołał tę funkcję, a rejestr a2 będzie zawierał wskaźnik do tablicy parametrów złożonej z długich słów zawierających następująco:

param[0] - parametr na pozycji zerowej odczytywany z (a2) zawiera typ requestera wywołującego daną funkcję użytkownika, dla funkcji rtFileRequestA() będzie się tam znajdowała wartość: REQHOOK_WILDFILE.

param[1] - parametr na pozycji pierwszej (drugie długie słowo) odczytywany z 4(a2) zawiera dla funkcji rtFileRequest() wskaźnik do struktury FileInfoBlock dla aktualnie wczytywanej pozycji katalogu.

Jeżeli funkcja użytkownika zwraca wartość FALSE (0) to nie zostają uwzględnione żadne zmiany, jeśli TRUE (1) to aktualna pozycja w katalogu, dla której została wywołana, funkcja użytkownika zostanie pominięta i nie pojawi się w katalogu.

Aby funkcja użytkownika była wołana dla każdego zbioru w katalogu należy ustawić znacznik FREQF_DOWILDFUNC.

Nasza struktura Hook powinna być zainicjowana następująco:

Hook-h_Entry - wpisujemy adres funkcji
(nie ma potrzeby inicjowania pola Hook_h_SubEntry
Hook-h_Data - dane użytkownika (jeżeli potrzebne)
następnie należy zainicjować pole w strukturze Req-Hook wpisując do niego wskaźnik naszej struktury Hook.

Zawsze należy sprawdzać dlaczego nasza funkcja została wywołana i zawsze należy zwracać odpowiednią wartość do requestera gdy kończymy jej wykonywanie. Wartość FALSE (0) została zarezerwowana na 'kontynuuj dalej'. Jeżeli nasza funkcja została wywołana z innego powodu niż się tego spodziewamy (typ niezrozumiały bądź nie obsługiwany) to powinniśmy zwrócić wartość zero!

Jeżeli zmieniamy procedurę obsługi bądź zachowanie sie istniejącej procedury to koniecznie (!!!) musimy wywołać funkcję rtFreeReqBuffer aby oczyścić bufor requestera.

Wejście:

Req - wskaźnik struktury requestera zainicjowanej przy pomocy funkcji rtAllocRequestA().

Nazwa - wskaźnik do bufora, w którym będzie umieszczona nazwa zbioru wybranego przez użytkownika (długość przynajmniej 108 bajtów).

Tytuł - wskaźnik do nazwy dla okna requestera (tekst musi być zakończony zerem).

TagList - wskaźnik listy rozkazów:

Dostępne rozkazy dla rtFileRequest().

RT_Window - jako parametr podajemy wskaźnik struktury Window dla okna, które będzie użyte aby odnaleźć ekran, na którym ma być wyświetlony requester. Jeżeli program jest zadaniem a nie procesem to musimy koniecznie podać ten rozkaz gdyż zadania w przeciwieństwie do procesów nie mają pr_WindowPtr.

RT_ReqPos - jako parametr podajemy długie słowo zawierające jeden z następujących znaczników:

REQPOS_POINTER - requester będzie się pojawiał tam gdzie znajduje się kursor myszki.

REQPOS_CENTERSCR - requester będzie centrowany na ekranie.

REQPOS_CENTERWINDOW - requester będzie centrowany w oknie (działa tylko wtedy jeżeli pr_WindowPtr jest właściwy lub gdy użyliśmy rozkazu RT_Window). Jeżeli RT_Window jest równe zero to requester bedzie centrowany na ekranie.

REQPOS_TOPLEFTSCR - requester będzie się pojawiał w lewym górnym rogu ekranu.

REQPOS_TOPLEFTWIN - requester będzie się pojawiał w lewym górnym rogu okna (jeżeli pr_WindowPtr jest właściwy lub gdy użyliśmy rozkazu RT_Window).

RT_LeftOffset - jako parametr podajemy długie słowo zawierające przesunięcie od lewej krawędzi wyznaczonej przez rozkaz RT_ReqPos (nie odnosi się to do requesterów, których miejsce pojawiania się zależne jest od pozycji kursora myszki to jest RT_ReqPos jest określone jako REQPOS_POINTER).

RT_TopOffset - jako parametr podajemy długie słowo zawierające przesuniecie od górnej krawędzi wyznaczonej przez rozkaz RT_ReqPos (nie odnosi się to do requesterów, których miejsce pojawiania się zależne jest od pozycji kursora myszki to jest RT_ReqPos jest określone jako REQPOS_POINTER).

RT_Screen - jako parametr podajemy wskaźnik struktury Screen dla ekranu, na którym ma się pojawić requester.

RT_ReqHandler - używając tego rozkazu możemy wystartować asynchroniczny requester. Jako parametr podajemy wskaźnik struktury rtHandlerInfo. Requester zainicjuje ten wskaźnik i powróci do programu po jego normalnej

AMIGA

inicjalizacji. Wartość zwrócona do programu nie jest taka jakiej moglibyśmy się spodziewać i jeżeli nie jest równa wartości CALL_HANDLER to znaczy, że wystąpił błąd i należy ustawiać parametry dla requestera od początku. Na temat obsługi takeigo requestera będzie mowa przy opisie funkcji rtReqHandlerA().

RT_WaitPointer - jako parametr podajemy wartość prawdziwą (TRUE) czyli większą od zera i - jeśli requester włączony - to kliknięcie myszką gdziekolwiek poza requesterem powoduje, że następuje zmiana wskaźnika myszki na wskaźnik uśpienia. Po wyjściu z requestera nasze okno nie będzie posiadało własnego wskaźnika (gdy mamy własny wskaźnik należy go ustawiać po wyjściu z requestera). Zalecane jest częste używanie tego rozkazu.

RT_DefaultFont - jako parametr podajemy wskaźnik struktury TextFont dla czcionki, która ma być użyta do wyświetlania tekstów w oknie requestera. Jeśli czcionka dla ekranu, na którym opojawia się requester, jest proporcjonalna to zostanie użyta czcionka standardowa pobierana z GfxBase-DefaultFont.

RTFI_Flags - jako parametr podajemy długie słowo zawierające znaczniki:

FREQF_NOBUFFER - przekazuje funkcji aby nie używała bufora do składowania zawartości katalogu. Następnym razem funkcja ma odczytywać katalog na nowo.

FREQF_MULTISELECT - włączenie tego znacznika umożliwia użytkownikowi wybór kilku zbiorów (zostanie wtedy uaktualniony gadżet do selekcji zbiorów) i można je wybierać poprzez trzymanie klawisza Shift i kliknięcia myszką na danym zbiorze. Procedura rtFileRequestA() zwróci wskaźnik do struktury rtFileList, która będzie zawierała wszystkie wybrane zbiory. Należy użyć procedury rt_FreeFileList() aby zwolnić pamięć przydzieloną na tę listę zbiorów.

FREQF_SELECTDIRS - ustawiamy ten znacznik jeżeli chcemy aby użytkownik mógł dokonywać wyboru zarówno katalogów jak i zbiorów Należy także ustawić znacznik FREQF_MULTISELECT. Katalogi będą zwracane razem ze zbiorami w liście zbiorów w RTFileList ale z wartością StrLen równą -1 Chcąc znać długość nazwy katalogu możemy użyć procedury strlen().

FREQF_SAVE - ten znacznik należy ustawić jeżeli requester jest wywoływany do nagrywania bądź kasowania zbiorów. Podwójne wciśnięcie przycisku myszki na nazwie zbioru będzie niedopuszczalne (użytkownikowi trudniej będzie popełnić pomyłkę i wybierając zły zbiór). Jeżeli ten znacznik jest ustawiony i użytkownik wpisze nazwę katalogu nie istniejącego to requester zwróci się z pytaniem czy ma utworzyć ten katalog.

FREQF_NOFILES - znacznik ustawiamy jeżeli chcemy aby requester umożliwiał użytkownikowi wybór jedynie katalogu. Zbiory nie będą wyświetlane. Można używać go razem ze znacznikami FREQF_MULTISELECT i FREQF-_SELECTDIRS.

FREQF_PATGAD - kiedy ten znacznik jest ustawiony to w oknie requestera pojawi się gadżet umożliwiający wpisanie do niego dowolnego ciągu znaków umożliwiającego selekcję zbiorów.

FREQF_DOWILDFUNC - jeżeli ten znacznik jest ustawiony do dla każdego zbioru będzie wywoływana funkcja (zdefiniowana przez programistę) służąca do selekcji zbiorów (opisane powyżej). Nie istnieje rozkaz ustawiający tę funkcję i należy jej adres wpisać do struktury Hook a adres tej struktury do struktury Req.

RTFI_Height - jako parametr podajemy wysokość okna requestera.

RTFI_OkText - jako parametr podajemy wskaźnik do tekstu zakończonego zerem (maksymalnie sześć liter), który będzie używany w requesterze zamiast tekstu 'Ok' w gadżecie do zatwierdzania zbioru. Jako tekst można wpisać np. (Load, Save, Delete. etc.)

Wyjście:

Wartość - wartość zwracana przez rtFileRequesA() w rejestrze D0. Jeżeli wartość ta odpowiada FALSE (0) to oznacza, iż użytkownik zaniechał wyboru zbioru, natomiast jeżeli odpowiada TRUE (1) to należy odczytać nazwę zbioru z bufora Nazwa a nazwę katalogu ze struktury Req z pola Dir (Req-Dir). Jeżeli mieliśmy ustawiony znacznik FREQF_MULTISELECT to w D0 otrzymamy wskaźnik do struktury rtFileList, z której pobieramy nazwy zbiorów, następnie musimy dokonać zwolnienia pamięci zajmowanego przez tę strukturę za pomocą rtFreFileList().

---

rtFreeFileList
rtFreeFileList (FileList)
A0

Procedura rtFreeFileList() zwalnia obszar pamięci przydzielony dla danych o zbiorach znajdujących się w strukturze FileList i umieszczonych tam przez procedurę rtFileRequestA() z ustawionym znacznikiem FREQF_MULTISELECT. Wywołujemy tę funkcję po tym jak przejrzeliśmy już listę zbiorów dostarczonych i nie potrzebujemy jej więcej.

Wejście:

FileList - wskaźnik do struktury rtFileList, zwracanej prez rtFileRequest (można podawać zero).

---

rtFontRequestA
Wartość = rtFontRequestA (Req, Tytuł, TagList)
D0 A1 A3 A0

Procedura rtFontRequestA() umożliwia użytkownikowi wybór czcionki i opcjonalnie stylu tej czcionki. Ustawiając odpowiednie znaczniki możemy osiągnąć efekt wywoływania naszej procedury po wczytaniu każdego ze zbiorów do katalogu. (Opis ustawienia takiej procedury znajduje się przy opisie funkcji rtFileRequestA()).

Tablica parametrow przekazywana do funkcji obsługi będzie posiadała jednak inne znaczenie a jej zawartość będzie się zdecydowanie różniła:

param[0] - parametr na pozycji zerowej odczytywany z (a2) zawiera typ requestera wywołującego daną funkcję użytkownika, dla funkcji rtFontRequestA() będzie się tam znajdowała wartość: REQHOOK_WILDFONT a nie - jak w przypadku procedury rtFileRequest - znacznik REQHOOK_WILDFILE.

param[1] - parametr na pozycji pierwszej (drugie długie słowo) odczytywany z 4(a2) zawiera dla funkcji rtFontRequest() wskaźnik do struktury TextAttr dla aktualnie wczytywanej czcionki do katalogu.

Jeżeli funkcja użytkownika zwraca wartość FALSE (0) to nie zostają uwzględnione żadne zmiany, jeśli TRUE (1) to aktualna czcionka w katalogu, dla której została wywołana

funkcja użytkownika, zostanie pominięta i nie pojawi się w katalogu.

Jeżeli zmieniamy procedurę obsługi bądź zachowanie sie istniejącej procedury to koniecznie (!!!) musimy wywołać funkcję rtFreeReqBuffer aby oczyścić bufor requestera.

Wejście:

Req - wskaźnik struktury requestera zainicjowanej przy pomocy funkcji rtAllocRequestA().

Tytuł - wskaźnik do nazwy dla okna requestera (tekst musi być zakończony zerem).

TagList - wskaźnik listy rozkazów:

Dostępne rozkazy dla rtFontRequest().

RT_Window - jako parametr podajemy wskaźnik struktury Window dla okna, które będzie użyte aby odnaleźć ekran, na którym ma być wyświetlony requester. Jeżeli program jest zadaniem a nie procesem to musimy koniecznie podać ten rozkaz gdyż zadania w przeciwieństwie do procesów nie mają pr_WindowPtr.

RT_ReqPos - jako parametr podajemy długie słowo zawierające jeden z następujących znaczników:

REQPOS_POINTER - requester będzie się pojawiał tam gdzie znajduje się kursor myszki.

REQPOS_CENTERSCR - requester będzie centrowany na ekranie.

REQPOS_CENTERWINDOW - requester będzie centrowany w oknie (działa tylko wtedy jeżeli pr_WindowPtr jest właściwy lub gdy użyliśmy rozkazu RT_Window). Jeżeli RT_Window jest równe zero to requester bedzie centrowany na ekranie.

REQPOS_TOPLEFTSCR - requester będzie się pojawiał w lewym górnym rogu ekranu.

REQPOS_TOPLEFTWIN - requester będzie się pojawiał w lewym górnym rogu okna (jeżeli pr_WindowPtr jest właściwy lub gdy użyliśmy rozkazu RT_Window)

RT_LeftOffset - jako parametr podajemy długie słowo zawierające przesuniecie od lewej krawędzi wyznaczonej przez rozkaz RT_ReqPos (nie odnosi się to do requesterów, które mają miejsce pojawiania się zależne od pozycji kursora myszki to jest RT_ReqPos jest określone jako REQPOS_POINTER).

RT_TopOffset - jako parametr podajemy długie słowo zawierające przesuniecie od górnej krawędzi wyznaczonej przez rozkaz RT_ReqPos (nie odnosi się to do requesterów, które mają miejsce pojawiania się zależne od pozycji kursora myszki to jest RT_ReqPos jest określone jako REQPOS_POINTER).

RT_Screen - jako parametr podajemy wskaźnik struktury Screen dla ekranu, na którym ma się pojawić requester.

RT_ReqHandler - używając tego rozkazu możemy wystartować asynchroniczny requester. Jako parametr podajemy wskaźnik struktury rtHandlerInfo. Requester zainicjuje ten wskaźnik i powróci do programu po jego normalnej inicjalizacji. Wartość zwrócona do programu nie jest taka jakiej moglibyśmy się spodziewwać, i jeżeli nie jest równa wartości CALL_HANDLER to znaczy, że wystąpił błąd i należy ustawiać parametry dla requestera od początku. Na temat obsługi takeigo requestera będzie mowa przy opisie funkcji rtReqHandlerA().

RT_WaitPointer - jako parametr podajemy wartość prawdziwą (TRUE) czyli większą od zera, jeśli requester jest włączony to kliknięcie myszką gdziekolwiek poza requesterem powoduje, że następuje zmiana wskaźnika myszki na wskaźnik uśpienia. Po wyjściu z requestera nasze okno nie będzie posiadało własnego wskaźnika (należy go ustawiać po wyjściu z requestera jeżeli mamy ustawiony ten rozkaz). Zalecane jest częste używanie tego rozkazu.

RT_DefaultFont - jako parametr podajemy wskaźnik struktury TextFont dla czcionki, która ma być użyta do wyświetlania tekstów w oknie requestera. Jeśli czcionka dla ekranu, na którym opojawia się requester jest proporcjonalna to zostanie użyta czcionka standardowa pobierana z GfxBase-DefaultFont.

RTFO_Flags - jako parametr podajemy długie słowo zawierające znaczniki:

FREQF_NOBUFFER - ustawiając ten znacznik likwidujemy buforowanie wczytywanych pozycji katalogu, co przy następnym wywołaniu procedury rtFontRequest spowoduje ponowne wczytywanie katalogu.

FREQF_FIXEDWIDTH - po ustawieniu tego znacznika w oknie requestera będą się pojawiały jedynie czcionki o ustalonej szerokości.

FREQF_COLORFONTS - jeżeli ustawimy ten znacznik to w oknie requesterą będą pojawiały się również czcionki kolorowe.

FREQF_CHANGEPALETTE - gdy ustawimy ten znacznik to procedura rtFontRequest dokona zmiany kolorów ekranu, na którym znajduje się okno requestera w ten sposób aby kolory ekranu odpowiadały kolorom czcionki kolorowej aktualnie wyświetlanej.

FREQF_LEAVEPALETTE - po ustawieniu tego znacznika procedura rtFontRequest() przy wyjściu będzie pozostawiała takie ustawienie kolorów jak przed wyjściem z tej procedury. Jest to bardzo użyteczne w połączeniu ze znacznikiem FREQF_CHANGEPALETTE.

FREQF_SCALE - znacznik zezwala na skalowanie czcionek.

FREQF_STYLE - ustawienie tego znacznika powoduje dołączenie gadżetów umożliwiających wybór stylu dla danej czcionki.

FREQF_DOWILDFUNC - jeżeli ten znacznik jest ustawiony do dla każdego zbioru będzie wywoływana funkcja (zdefiniowana przez programistę) służąca do selekcji zbiorów (opisane przy procedurze rtFileRequestA()). Nie istnieje rozkaz ustawiający tę funkcję i należy jej adres wpisać do struktury Hook a adres tej struktury do struktury Req.

RTFO_Height - jako parametr podajemy wysokość okna requestera.

RTFO_OkText - jako parametr podajemy wskaźnik do tekstu zakończonego zerem (maksymalnie sześć liter), który będzie używany w requesterze zamiast tekstu 'Ok' w gadżecie do zatwierdzania zbioru. Jako tekst można wpisać np. (Load, Save, Delete. etc.)

RTFO_sampleHeight - jako parametr podajemy wysokość obszaru, w którym procedura rtFontRequestA() będzie nam przedstawiała próbki danej czcionki. Jeżeli nie ustawimy tego rozkazu to zostaje przyjęta wartość 24.

RTFO_MinHeight - jako parametr podajemy minimalną wielkość czcionki, która będzie wyświetlana.

RTFO_MaxHeight - jako parametr podajemy maksymalną wielkość czcionki, która będzie wyświetlana.

Wyjście:

Wartość - jeżeli użytkownik dokonał wyboru czcionki to procedura rtFontRequestA() zwróci wartość TRUE (1), w przypadku gdy użytkownik zaniechał wyboru otrzymamy wartość FALSE (0).

AMIGA

Struktury dla procedur rtFileRequestA(), rtFontRequesta() i rtFreeFileList().

Struktura rtFileRequester musi być przydzielona za pomocą procedury rtAllocRequestA():

STRUCTURE rtFileRequester

00 ULONG rtfi_ReqPos - znaczniki ReqPos.

04 UWORD rtfi_LeftOffset - pozycja okna requestera (lewa krawędź).

06 UWORD rtfi_TopOffset - pozycja okna requestera (górna krawędź).

08 ULONG rtfi_Flags - znaczniki dotyczące zachowania się requestera.

0C APTR rtfi_Hook - adres funkcji użytkownika wywoływanej do selekcji zbiorów.

10 APTR rtfi_Dir - wskaźnik do nazwy katalogu, z którego został wybrany zbiór. (Pole tylko odczytywalne. Zmian można dokonywać jedynie za pomocą procedury rt ChangeReqAttrA()).

14 APTR rtfi_MatchPat - wskaźnik do tekstu służącego do wyłączania zbiorów z listy katalogu. (Pole tylko odczytywalne. Zmian można dokonywać jedynie za pomocą procedury rt ChangeReqAttrA()).

18 APTR rtfi_DefaultFont - wskaźnik czcionki używanej dla danego requestera.

1C ULONG rtfi_WaitPointer - wskaźnik do danych opisujących wygląd kursora myszki podczas pracy requestera.

Dalej następują dane tylko dostępne dla biblioteki ReqTools.library i należy uważać aby ich nie uszkodzić.

Struktura rtFontRequester tak samo jak rtFileRequester musi być przydzielona za pomocą procedury rtAllocRequestA():

STRUCTURE rtFontRequester

00 ULONG rtfo_ReqPos - znaczniki ReqPos.

04 UWORD rtfo_LeftOffset - pozycja okna requestera (lewa krawędź).

06 UWORD rtfo_TopOffset - pozycja okna requestera (górna krawędź).

08 ULONG rtfo_Flags - znaczniki dotyczące zachowania się requestera.

0C APTR rtfi_Hook - adres funkcji użytkownika wywoływanej do selekcji zbiorów.

10 STRUCT rtfo_Attr[8] - struktura TextAttr dla aktualnie wczytywanej czcionki

18 APTR rtfo_DefaultFont - wskaźnik czcionki używanej dla danego requestera.

1C ULONG rtfo_WaitPointer - wskaźnik do danych opisujących wygląd kursora myszki podczas pracy requestera.

Struktura rtFileList jest zwracana przez procedurę rtFileRequestA() jeżeli został ustawiony znacznik FREQF_MULTISELECT.

STRUCTURE rtFileList

00 APTR rtfl_Next ; wskaźnik do następnej struktury FileList

04 ULONG rtfl_StrLen ; długość nazwy zbioru lub -1 gdy zamiast nazwy zbioru występuje nazwa katalogu.

08 APTR rtfl_Name ; wskaźnik do nazwy zbioru bądź katalogu.

Struktura Hook jest inicjowana dla naszych własnych funkcji i inicjujemy ją w następujący sposób:

STRUCTURE Hook

00 STRUCT MLN[8] - miejsce zarezerwowane na strukturę MLN (rezerwujemy 8 bajtów - dwa słowa na MLN_SUCC i MLN_PRED i ustawiamy na zero).

08 APTR h_Entry - adres naszej procedury.

0C APTR h_SubEntry - ustawiamy na zero.

10 APTR h_Data - pole na adres dla danych użytkownika.

I na zakończenie jeszcze jedna procedura zawarta w bibliotece ReqTools.Library:

rtSetReqPosition

rtSetReqPosition (ReqPos, NewWindow, Screen, Window)

D0 A0 A1 A2

Procedura rtSetReqPosition ustala położenie lewego górnego rogu okna (w strukturze NewWindow) względem znaczników zawartych w rejestrze D0 (ReqPos).

Struktura NewWindow musi być zainicjowana przed wejściem do tej procedury.

Pola NewWindow-LeftEdge (lewa krawędź) i NewWindow-TopEdge (górna krawędź) będą używane jako przesunięcie przy niektórych znacznikach ReqPos (znaczniki: REQPOS_TOPLEFTSCR i REQPOS_TOPLEFTWIN). Jeżeli chcemy mieć okno na pozycji (10,40) względem lewego górnego rogu ekranu to w pole NewWindow-LeftEdge wpisujemy wartość 10 a w pole NewWindow-TopEdge wpisujemy wartość 40 i wywołujemy procedurę rtSetReqPosition ze znacznikiem REQPOS_TOPLEFTSCR. Jeżeli nie chcemy przesuniętego okna to należy pamiętać o tym aby w pole NewWindow-LeftEdge i w pole NewWindow-TopEdge wpisać wartość 0. W przypadku użycia znacznika REQPOS_POINTER (czyli ustawianie okna względem kursora myszki) do tych pól należy wpisać wartości odpowiadające punktowi który wskazywać ma kursor w tym oknie po jego otworzeniu. Jeżeli w pole NewWindow-LeftEdge i w NewWindow-TopEdge wpiszemy wartość 0 i wywołamy funkcję rtSetReqPosition() to otrzymamy okno, którego lewy górny róg będzie znajdował się w pozycji kursora. Wskaźnik struktury Screen może być niezerowy ale dla okna otwieranego należy wtedy wpisać adres tej struktury do pola NewScreen-WScreen. Wskaźnik struktury okna jest jedynie potrzebny dla okien z ustawionym znacznikiem REQPOS_CENTERWIN i REQPOS_TOPLEFTWIN. W przypadku ustawienia tych znaczników i nie podania wskaźnika struktury Window (Window=0) możemy spokojnie wywoływać procedurę rtSetReqPosition gdyż znaczniki automatycznie zostaną zamienione na REQPOS_CENTERSCR i REQPOS_TOPLEFTSCR.

Wejście:

ReqPos - jeden ze znaczników dotyczących ustawiania pozycji requestera (REQPOS_...).

NewWindow - wskaźnik struktury NewWindow (całkowicie zainicjowanej).

Screen - wskaźnik struktury Screen dla ekranu na którym ma pojawić się requester.

Window - wskaźnik struktury Window (jeżeli nasz requester ma się pojawić w obrębie okna) lub zero.

Marcin "Duddie" Dudar

MONKEY ISLAND II

AMIGA

W poprzednim odcinku instrukcji opisaliśmy dwie pierwsze części tej wspaniałej gry. Dzisiaj zakończymy nasze przygody...

Część trzecia: "LeChuck's Fortress"

Zaczynamy we wnętrzu fortecy. Stoimy obok pudła, w którym odbyliśmy naszą jakże wyczerpującą podróż do frotecy LeChucka. Idziemy w prawo, a następnie do góry, po schodach aż dojdziemy do drzwi. Po wejściu musimy iść prawym korytarzem. Będziemy przechodzić obok tablicy z drogowskazami, obok dziwnych szkieletów - nie zwracamy jednak uwagi na te komnaty. Na końcu korytarza wchodzimy w drzwi i po drugiej ich stronie schodzimy po schodach na dół, do lochów. Na dole, gdy podejdziemy do celi zauważymy Wallego (naszego znajomego kartografa) wiszącego na łańcuchach. Porozmawiamy z nim trochę próbując podtrzymać go na duchu. Po skończonej romowie wracamy do komnaty z dziwnymi szkieletami powieszonymi na ścianach (Ugly Bone Things). Czy nam to niczego nie przypomina? Oczywiście, że tak! Szkielety te są podobne do tych, o których mowa w piosence naszych rodziców (której słowa mamy zapisane na kartce). Bierzemy kartkę i czytamy pierwszy jej werset. Jeżeli przed sobą nie widzimy podobnej konfiguracji to musimy iść dalej w lewo, aż dojdziemy do komnaty, w której będzie się znajdował opisany w piosence szkielet. Gdy go znajdziemy popchnijmy wnękę z tymże szkieletem i odkryjemy tajne przejście do następnej komnaty.

Tutaj komnacie musimy ponownie zajrzeć do naszych notatek. Czytamy drugi wers piosenki i znajdujemy odpowiedni szkielet. Czynność tą musimy powtarzać aż cztery razy.

Po otwarciu ostatnich drzwi trafimy do dość dużej komnaty z ogromnymi drzwiamy zamkniętymi na wiele różnej wielkości zamków. Nie dajmy się jednak zwieść pozorom i po prostu otwieramy drzwi. Wchodzimy do następnej komnaty. Znajduje się w niej tron LeChucka, obok niego wisi klucz. Podejdź i weź klucz. W tym momencie będziemy złapani w pułapkę LeChucka. Po krótkiej animacji znajdujemy się w lochu przywiązani sznurami do sufitu. Wisimy sobie nad naczyniem zawierającym kwas i czekamy aż po wypaleniu się świeczki cała maszyneria ruszy i spowoduje, że znajdziemy się w naczyniu. Działanie wszystkich urządzeń wyjaśni nam sam LeChuck. Możemy mu jeszcze zadać kilka głupich pytań, aż w końcu znudzony opuści nas. Teraz możemy zabrać się do roboty. Pierwsza czynnością, jaką należy zrobić będzie oczywiście zgaszenie świeczki. W tym celu najpierw musimy napić się jednego z naszych drinków, oczywiście przy pomocy słomki.

Rozejrzyjmy się teraz po pomieszczeniu. Z tego, co mówił LeChuck, możemy zgasić świeczkę korzystając z tarczy znajdującej się po naszej prawej stronie. Spróbujmy splunąć w lewy dolny róg tej tarczy, gdy dobrze wcelujemy nasza ślina po krótkim locie powinna zgasić świeczkę. Jeżeli nie uda się to za pierwszym razem możemy próbować aż do osiągnięcia pełnego sukcesu.

Po zgaszeniu świeczki zrobi się trochę zamieszania. Gdy wszystko się uspokoi znajdziemy się w ciemnym pomieszczeniu. Aby zobaczyć co się w nim znajduje użyjemy oczywiście zapałek. Zapalamy zapałki i...

Część czwarta: "Dinky Island"

Tak oto doszliśmy do ostatniej części gry. Na dojście do jej zaskakującego końca musimy jednak jeszcze trochę poczekać.

Na początku przybywamy na wyspę Dinky. Idziemy w prawo i spotykamy tam Hermana Toothrota. Możemy z nim porozmawiać (ale nie musimy). Na jego pytane ile kolorów ma drzewo, które mamy przed sobą odpowiemy, że ma wszystkie kolory. Herman zgodzi się z nami i chwilowo nas opuści. Teraz idziemy w lewo. Powinniśmy znaleźć tu szklaneczkę oraz łom. Weźmy obie rzeczy. Popatrzmy teraz na lewy dolny róg ekranu. Częściowo w wodzie, a częściowo na lądzie leży tam butelka. Bierzemy ją i podchodzimy do leżącego obok kamienia. Teraz możemy otworzyć butelkę. Wystarczy zadziałać butelką na kamień (USE BOTTLE ON ROCK). Idziemy teraz do beczki ustawionej zaraz pod papugą. Warto zobaczyć, co się w niej znajduje, tak więc otwórzmy ją. Po chwili mamy już ciasteczko, które z pewnością zasmakuje papudze (my nie mamy ochoty na słodycze). Dajemy więc je papudze i w zamian uzyskujemy informację o miejscu, w którym znajduje się skarb. Następnie podchodzimy do brzegu i napełnimy naszą szklaneczkę morską wodą (USE MARTINI GLASS ON OCEAN). Teraz musimy jeszcze otrzymać ze słonej wody morskiej wodę słodką, najlepiej destylowaną. Wiemy już co zrobić? (USE MARTINI GLASS WITH STILL) Mamy już odpowiednią wodę, możemy więc działać dalej.

Idziemy w lewo, aż dojdziemy do dziury prowadzącej do dżungli. Oczywiście korzystamy z możliwości wejścia do lasu tropikalnego. W dżungli idziemy ścieżką do góry, a następnie w lewo.

Teraz idziemy prosto aż do następnej ścieżki w lewo, tam skręcamy i kierujemy się prosto do góry, gdzie zobaczymy drzewo, na którym wisi torba. Odcinamy ją. W tym celu posłużymy się wcześniej znalezioną butelką (USE BROKEN BOTTLE ON BAG). Z torby wypadnie na ziemię pudełko krakersów. Bierzemy je ze sobą. Możemy teraz już wyjść z dżungli. Wracamy tą samą drogą.

Gdy znajdziemy się ponownie przy wejściu do dżungli idziemy ścieżką kierującą się w górę ekranu. Na miejscu znajdziemy pudełko zawiązane sznurkiem. Aby je zdobyć

musisz zdjąć sznurek. Następnie, aby je otworzyć trzeba będzie posłużyć się łomem (USE CROWBAR ON BOX). Gdy mamy już odpowiednie materiały wybuchowe możemy udać się w dalszą drogę.

Idziemy na wschód (to znaczy w prawo) i po kilu ekranach znajdziemy się na planszy z dinozaurem. Gdy już będziemy na miejscu po chwili znajdzie się tutaj także nasza znajoma papuga. Musisz teraz przygotować krakersy - korzystasz z wody (USE MARTINI GLASS WITH WATER ON CRACKER MIX). W ten sposób zamiast jednego otrzymamy dwa krakersy. Jeden z nich musimy dać papudze - za co otrzymamy kolejną wskazówkę.

Idziemy na północ (zgodnie ze wskazówką papugi) i w końcu docieramy do kupki kamieni. Tutaj także po chwili nadlatuje nasz nieodłączny przyjaciel. Nie pozostaje nam oczywiście nic innego, jak tylko dać mu naszego ostaniego krakersa. W zamian za takie poświęcenie otrzymujemy ostatnią wskazówke potrzebną do znalezienia poszukiwanego skarbu. Oczywiście postępujemy zgodnie z podaną informacją i po krótkim marszu na wschód (oczywiście idziemy w prawo) docieramy na miejsce zaznaczone dużym 'X'. Niestety, w tym momencie Herman musi nam zaprezentować swoją wiedzę o wyspie i pokazać swoją drogę do 'X'. Wracamy z nim do obozu, następnie wracamy do naszego dużego 'X' drogą wskazaną przez Hermana. Będąc ponownie przy znaku zaczniemy oczywiście kopać. Bierzemy łopatę i zabieramy się do roboty (USE SHOVEL ON BIG X).

Teraz, gdy mamy już wykopaną odpowiedniej wielkości dziurę użyjemy dynamitu: Najpierw trzeba zapalić nasze ładunki (USE MATCHES ON DYNAMITE) i następnie wstawić je do dziury (USE LIT DYNAMITE ON HOLE). To tyle...

Po wielkim wybuchu znajdziemy się w środku dziury na linie (czy to nam czegoś przypadkiem nie przypomina?). Aby teraz nie bujać się bezmyślnie na linie usztywnimy nieco połączenie za pomocą łomu (USE ROPE WITH CROWBAR oraz USE ROPE & CROWBAR WITH TWISTED RODS). Po krótkiej rozmowie z Elaine lina niestety się urywa i spadamy w głęboką ciemność. Osiągnąwszy dno spróbujmy znaleźć włącznik światła i użyjmy go. Może jednak nie powinniśmy tego robić? Cóż, stało się. Możemy teraz odbyć rozmowę z LeChuckiem.

Po chwili użyje on lalki voodoo, co spowoduje pewne zmiany w naszym położeniu. Wracamy teraz do pokoju, w którym znaleźliśmy skarb. Weźmy ze skrzyni po skarbie znajdujący się w niej bilet. Teraz idziemy w prawo na sam koniec tunelu. Znajdźmy drzwi do gabinetu pierwszej pomocy. Oczywiście musimy je otworzyć, po czym wchodzimy do środka. Rozejrzyjmy się uważnie po pokoju. Zwłaszcza warto zwrócić uwagę na stojące tu szkielety. Bierzemy czaszkę tatusia (PICK UP DADS SKULL). Podejdźmy teraz do kubła na śmieci znajdującego się pod znakiem pierwszej pomocy. Otwieramy wiadro, po czym wyjmujemy z niego rękawice chirurgiczne. Spójrzmy jeszcze na szufladę.

Gdy załatwimy już wszystkie sprawy możemy opuścić pokój i udać się w lewą stronę tunelem. Wchodzimy w pierwsze drzwi (po drzwiach do gabinetu pierwszej pomocy). Znajdziemy się w magazynie. Tutaj musimy pootwierać wszystkie pudełka znajdujące się na ziemi. Musimy też znaleźć i wziąć następujące rzeczy: balon, piwo korzenne oraz lalkę voodoo. I to już wszystko, co mieliśmy do zrobienia w tym pokoju. Opuszczamy go więc i idziemy dalej tunelem w lewą stronę.

Kolejne drzwi znajdują się na prawo od znaku oznaczającego windę. Oczywiście wchodzimy także do tego pokoju. Znajduje się tu zepsuta maszyna do produkowania grogu. Użyjmy rękawic chirurgicznych i balonu z helem. Oglądając maszynę do produkcji grogu zauważymy otwór zwracający monety. Spróbujmy coś z nim zrobić (USE COIN RETURN). W tym momencie jedna moneta wypadnie na podłogę. Po chwili LeChuck powinien wejść i podnieść monetę z podłogi. Gdy nie przychodzi to poczekaj chwilę lub powtórz cały manewr z monetą. Kiedy LeChuck schyli się po monetę będziemy mieli szansę oderwać kawałek jego ubrania (PULL UNDERWEAR). LeChuck użyje na tobie lalki voodoo co spowoduje przeniesienie cię do innego pokoju. Oczywiście wracamy do pokoju z maszyną do produkcji grogu. Gdy tym razem zrobimy naszą sztuczkę z monetą i LeChuck wejdzie ją podnieść musimy postarać się o kolejny składnik potrzebny do stworzenia lalki voodoo. Dajmy staremu piratowi wydmuchać nos (USE HANKIE WITH LECHUCK). Po wykonaniu czynności LeChuck ponownie przeniesie nas do innego pokoju. Tym razem nie wracamy już do pokoju z maszyną, ale idziemy do windy. Naciskamy guzik wzywający kabinę i po chwili możemy wejść do środka.

Stajemy przy dźwigni służącej do uruchomienia windy i gdy na horyzoncie pojawi się LeChuck czakamy na odpowiedni moment, aby zamknąć mu drzwi przed nosem. Dzięki temu możemy mu odpowienio przyciąć brodę. Winda pojedzie do góry. Gdy znajdziemy się na miejscu otwórzmy drzwi i wyjdźmy. Możemy teraz spokojnie przystąpić do pracy nad stworzeniem lalki voodoo działającej na LeChucka. Aby ją zmontować musimy do torby, którą otrzymaliśmy od kapłanki wrzucić wszystkie zebrane składniki, a więc lalkę, czaszkę, chustkę, brodę i fragment ubrania (DOLL, SKULL, BEARD, HANKIE, UNDERWEAR). Po wrzuceniu ostatniej z rzeczy potrząśniemy jeszcze torbą i mamy gotową lalkę voodoo, czyli całkiem niezłą broń przeciwko LeChuckowi.

Teraz ponownie idziemy do windy i zjeżdżamy na dół. Udajemy się na poszukiwania LeChucka. Nie powinny one długo trwać, więc wkrótce staniemy z naszym śmiertelnym wrogiem twarzą w twarz. W tym momencie musimy szybko wyjąć naszą lalkę voodoo i użyć jej. LeChuck coś powie i wyjdzie z pokoju. Idziemy za nim do następnego pokoju. Tam ponownie odbywamy krótką rozmowę. Wybierzmy opcję dialogową mówiącą o nodze, którą możemy oderwać od lalki voodoo... Po tym stwierdzeniu LeChuck jeszcze z nami trochę porozmawia, a następnie poprosi nas, abyśmy zdjęli maskę z jego twarzy. Oczywiście zgodzimy się i wykonamy jego prośbę. Gdy już to zrobimy... skończyliśmy ostanią część gry MONKEY ISLAND !!!!!

Od tego momentu możemy obejrzeć sekwencję na zakończenie, a następnie wyłączyć komputer i czekać na trzecią część tej wspaniałej gry.

Dla miłośników gierek adventure mam jednak dobrą wiadomość! Prawdopodobnie wkrótce ukaże się na Amidze (lub już się ukazała!) kolejna gierka z serii Indiana Jones. Będzie to już czwarta część i jest podobno wspaniała!

Jarosław "Jarri" Horodecki